WL-TR-97-2046

PROBABILISTIC ROTOR DESIGN SYSTEM (PRDS) PHASE 2 INTERIM REPORT

P. G. ROTH
Technical Program Manager

GE AIRCRAFT ENGINES
ADVANCED ENGINEERING PROGRAMS DEPARTMENT
ONE NEUMANN WAY
CINCINNATI, OHIO 54215-6301

JULY 1996

INTERIM REPORT THE FOR PERIOD 1 SEP 89 - 30 JUL 96

AERO PROPULSION AND POWER DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7251

NOTICE

WHEN GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA ARE USED FOR ANY PURPOSE OTHER THAN IN CONNECTION WITH A DEFINITELY GOVERNMENT-RELATED PROCUREMENT, THE UNITED STATES GOVERNMENT INCURS NO RESPONSIBILITY OR ANY OBLIGATION WHATSOEVER. THE FACT THAT THE GOVERNMENT MAY HAVE FORMULATED OR IN ANY WAY SUPPLIED THE SAID DRAWINGS, SPECIFICATIONS, OR OTHER DATA, IS NOT TO BE REGARDED BY IMPLICATION, OR OTHERWISE IN ANY MANNER CONSTRUED, AS LICENSING THE HOLDER, OR ANY OTHER PERSON OR CORPORATION; OR AS CONVEYING ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE OR SELL ANY PATENTED INVENTION THAT MAY IN ANY WAY BE RELATED THERETO.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

Theodore G. Fecke

THEODORE G. FECKE
CHIEF, Components Branch
Turbine Engine Division
Aero Propulsion & Power Directorate

Richard J. Hill

RICHARD J. HILL
Chief of Technology
Turbine Engine Division
Aero Propulsion & Power Directorate

IF YOUR ADDRESS HAS CHANGED, IF YOU WISH TO BE REMOVED FROM OUR MAILING LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION, PLEASE NOTIFY WL/POTC, WRIGHT-PATTERSON AFB, OH 45433-7251 TO HELP MAINTAIN A CURRENT MAILING LIST.

COPIES OF THIS REPORT SHOULD NOT BE RETURNED UNLESS RETURN IS REQUIRED BY SECURITY CONSIDERATIONS, CONTRACTUAL OBLIGATIONS, OR NOTICE ON A SPECIFIC DOCUMENT.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*

2. REPORT DATE
July 30, 1996

3. REPORT TYPE AND DATES COVERED
Interim report for period 1 Sep 89 - 30 Jul 96

4. TITLE AND SUBTITLE
Probabilistic Rotor Design System (PRDS) Phase II Interim Report

5. FUNDING NUMBERS
F33615-90-C-2070
PE 62203
PR 3066
TA 12
WU 80

6. AUTHOR(S)
P.G. Roth, Technical Program Manager

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
GE Aircraft Engines, Advanced Engineering Programs Department
One Neumann Way
Cincinnati, OH 54215-6301

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AERO PROPULSION AND POWER DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433- 7251
POC: Sqn Ldr Andrew J. Green, WL/POTC, 937-255-2734

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TR-97-2046

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE,
DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The USAF Probabilistic Rotor Design System (PRDS) contract was designed to develop, validate and demonstrate a probabilistic alternative to existing deterministic design philosophy. This report documents Phase II effort: Method Development. It begins with an introduction to the concept of designing probabilistically against failure and a comparison of three standard approaches to estimating failure probability. Following this, the philosophy is laid out for PDAS (PDAS — Probabilistic Design Analysis System), a software package which facilitates application to the design process of probabilistic algorithms and optimization tools. A general strategy is outlined for application of probabilistics with three specific examples provided to demonstrate the integration of statistically describable variability into design analyses. Specific proposals are then made for setting part lives using probabilistics. The report concludes with a detailed summary of the PDAS Template Language and its application to managing response surface analyses.

14. SUBJECT TERMS
Gas turbine engine design, Probabilistic methods, Risk analysis, Risk management

15. NUMBER OF PAGES
53

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18298-102

Table of Contents

Table of Contents

Table of Contents

List of Illustrations

List of Tables

1.0 Introduction and Summary

The USAF Probabilistic Rotor Design System (PRDS) contract (F33615–90–C–2070) was designed to develop, validate, and demonstrate a probabilistic alternative to existing deterministic design philosophy. The full statement of work is reproduced in Appendix A.

The Phase I (Data Acquisition) Interim Report (Reference 1) documented the following:

- Compilation of data from FAA failure statistics, GEAE cracking investigations and laboratory method validations. This data was used both to support analysis of the effectiveness of current design practices and analytical tools, and as background for a definition of acceptable risk.
- Review of GEAE cracking investigations to identify occasions of excessive stress or temperature, poor material properties, inherent material inclusions, manufacturing damage and so forth, to assess the effectiveness of current design practices which are largely based on safety factors.
- Analysis of laboratory data to assess the accuracy of specific analytic tools used during the design process.
- Identification of the principal failure modes to be incorporated into the probabilistic design analysis system and listing of the potential drivers or failure causes (which include stresses, temperatures and gradients).
- Algorithmic development and program flowcharting addressing four principal failure modes. As part of this development, relevant parameters such as part dimensions, inclusion sizes, speeds, temperatures and so forth were identified, and the approaches to be taken to integrate variations in these parameters into estimated risk outlined.
- Utilization of the FAA statistics gathered in Task 1 to draw conclusions concerning risk of disk failure in today's commercial engine fleet. It was assumed that design practices and analytical tools are largely the same regardless of the application (i.e. military or commercial), and hence that conclusions drawn from the FAA data are broadly applicable. Data was sorted by failure cause at the component level, and some trends were observed. Failure of turbine disks is more likely than failure of compressor disks is more likely than failure of fan disks. Disk failures are most often not design related.
- Developed a rationale for the application of probabilistic design and used the FAA data to suggest an acceptable level of risk for design.

This report documents Phase II effort – Method Development. It begins with an introduction to the concept of designing probabilistically against failure and a comparison of three standard approaches to estimating failure probability. Following this, the philosophy is laid out for PDAS, a software package which facilitates application to the design process of probabilistic algorithms and optimization tools.

The development of PDAS was motivated by and initiated under this contract. It is built on preexisting probabilistic fracture mechanics software developed at GEAE to address inclusion initiated cracking in fracture critical components. Fatigue, burst, and plastic deformation failure modes were addressed in principle in the Phase I Interim report, and PDAS has been applied at GEAE for probabilistic analysis of HCF and creep in blades. These failure modes were not evident, however, in the testing of Phase III of the PRDS program – Method Validation.

A general strategy is outlined for application of probabilistics with three specific examples provided to demonstrate the integration of statistically describable variability into design analyses. Specific proposals are then made for setting part lives using probabilistics. The report concludes with a detailed summary of the PDAS template language and its application to managing response surface analyses.

Acknowledgements

I wish to acknowledge the work of John Coomer and Norm Austin who shared technical responsibility for this contract for its first several years. Also, Don Beeson, Greg Blanc, Bob Maffeo, Bob McClain, Steve Schrantz, J.R. Engebretsen and Tim Stone for contributions to the development of PDAS. Finally, I wish to thank my USAF sponsors and contract monitors for their support: Ted Fecke, Mike McIntyre, Chris Pomfret, and Andrew Green.

Disclaimer

The examples developed in this report have been constructed for purpose of demonstration. While some data may be real, the applications do not represent hardware produced by GE Aircraft Engines.

2.0 Introduction to Probabilistic Design Analysis

Science (hence, science-based engineering) builds upon the conviction that the world works by cause and effect; actions are predictable functions of initial conditions given the right equations and adequate computational power.

If inputs can be accurately measured, and outcomes are not overly sensitive to allowable deviations, this ideal is achievable. A gearbox predictably converts rotation of the input shaft into rotation of the output shaft. A hydraulic actuator translates electric potential into ram force with relatively little deviation from a governing empirical equation.

Often, however, it is the case that controlling parameters are poorly characterized and widely variable, and/or that response is sensitive to small parameter changes, and/or that response exhibits a degree of randomness which cannot be tied to identified parameters. Fatigue life is correlated to grain size, but grain size is only an approximate measure and the correlation is not perfect – scatter is observed. Cause and effect works on average, but the deviations can not be dismissed.

Variability is an annoying fact of life for all designers and manufacturers. Products from ball bats to bridges do not always perform as advertised with failures ranging from partial and of little consequence to total and catastrophic. The options available to prevent failures are sometimes limited, may be counterproductive and in some cases may simply not be worth the trouble.

Baseball bats sometimes crack; the result is most often not life–threatening and, therefore, little has been done to redesign the concept to reduce the occurrence. Bridges sometimes collapse; the result is often loss of life and usually disruption of commerce and mobility. Civil engineering practices consequently opt for overdesign of such structures. Critical support elements may be sized to handle two or three times the maximum foreseeable service load.

There are situations when overdesign is not practical, for example where added bulk defeats the purpose of a system. Excess weight hinders performance in flight for birds, for aircraft, and for space systems. Aerodynamic forces of weight and drag can be overcome by thrust, but thrust requires energy, depletion of natural resources, and increase in pollution. In military applications, increasing weight decreases survivability when maneuverability is critical.

A design may be adjusted by rules of thumb and the risk of failure subsequently ignored as extremely (or, at least, adequately) remote, or the likelihood of failure may be recognized, quantified and controlled. Probabilistic design analysis methods attempt the latter. In the simplest case:

1. System response is completely determined by parameters $p_1, p_2, \ldots, p_n$ which are statistically distributed with distribution $D(p_1, p_2, \ldots, p_n)$.
2. The response can be quantified as a function of the parameters: $R(p_1, p_2, \ldots, p_n)$.
3. A failure region can be specified: $\{p_1, p_2, \ldots, p_n : R(p_1, p_2, \ldots, p_n) \text{ represents failure}\}$.

In theory, this information is sufficient to determine a failure probability, a measure of risk. The design can be adjusted and/or the parameter distributions controlled to hold failure probability below an appropriate limit.

Much published research focuses on tools used to estimate failure probability (see Reference 2 for a summary of current approaches). In the next two sections, three algorithms will be evaluated: Direct Numerical Integration, Monte Carlo Integration, and the First Order Reliability Approximation.

2.1 An Example

The following scenario, while admittedly contrived, is easily stated yet challenging.

Problem Statement

Assume that there are identical pressure waves incident on a target produced by two simultaneous detonations separated by a distance 2v. The midpoint of the detonations is related to the target by coordinates (x, y) relative to axes centered at the midpoint (Figure 1). Targeting is not precise. The point (x, y) is symmetrically binormally distributed about a nominal center (x_0, y_0) with standard deviation σ. Failure of the site occurs if the total pressure exceeds an ultimate value ρ. What is the failure probability?

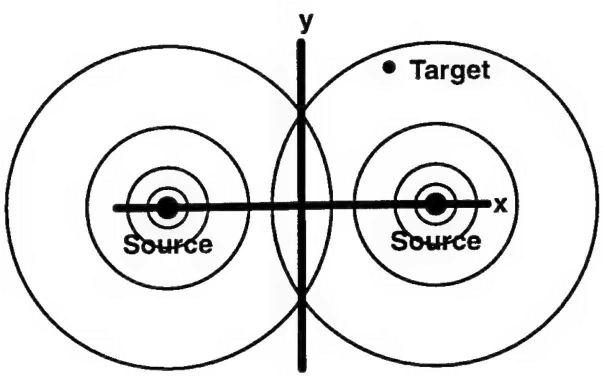
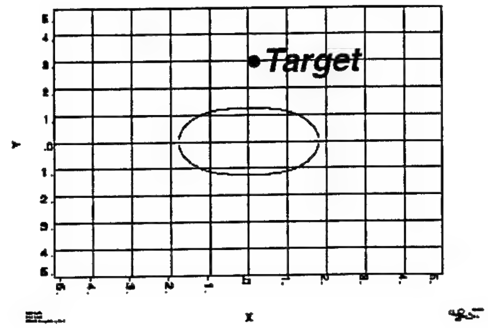


Figure 1. Dual Pressure Waves Hitting a Target.

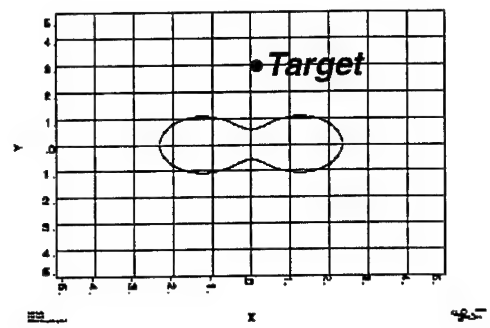
Peak pressure from each detonation varies inversely with the square of the distance from the center. Total peak pressure will be approximated by the sum of the individual peaks:

$$P(x, y) = p \cdot [(x - v)^2 + y^2]^{-1} + p \cdot [(x + v)^2 + y^2]^{-1}$$

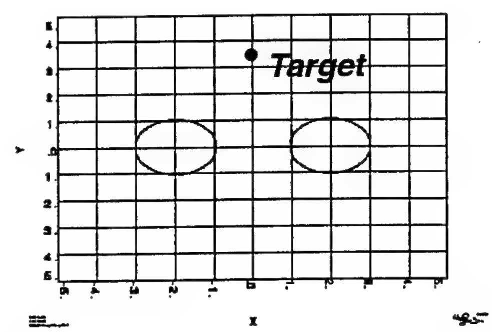
The failure region $\{(x, y) : P(x, y) > \rho\}$ can have a variety of shapes as shown in Figure 2 for the specific choices $(x_0, y_0) = (0, 3)$, $\rho = 1$ and three values of v: 0.7, 1.3 and 2.0. With σ set to 1, estimates are made of the probability F associated with each failure region:



v = 0.7



v = 1.3



v = 2.0

Figure 2. Failure Regions.

Direct Numerical Integration

$$F = \frac{1}{2}\pi \int \exp\left[\frac{(x-x_0)^2}{2\sigma^2}\right] \int_{y_{min}(x)}^{y_{max}(x)} \exp\left[\frac{(y-y_0)^2}{2\sigma^2}\right] dydx$$

where:
$y_{min}(x)$ and $y_{max}(x)$ solve the equation $P(x,y)=\rho$

Monte Carlo Integration

A random sample of points is generated from the binormal distribution: $\{(x_i, y_i) : i=1,10^6\}$.

F=fraction of points satisfying $P(x_i, y_i) > \rho$

First Order Reliability Approximation

The failure regions are simply bounded as in Figure 3.

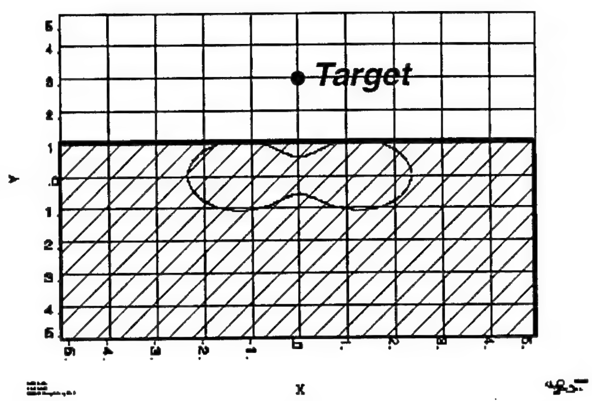


Figure 3. First Order Approximation to Failure Region.

$F=N(-D/\sigma)$

where: D is the distance of the point (x_0, y_0) from the shaded half–plane, and N(z) is the standard normal distribution.

The results are compared in Table 1. Direct numerical integration and Monte Carlo integration yield essentially identical estimates for all three values of v; the first order reliability method, as implemented, is uniformly conservative. The latter estimate could be improved by choosing different bounding regions (e.g. the polyhedral region shown in Figure 4), but evaluation of the probabilities associated with the more complex regions is generally no easier than executing the exact numerical integrations.

Table 1. Failure Results Comparison.

V	Direct Numerical Integration	Monte Carlo Integration	First Order Reliability Method
0.7	0.0295	0.0297	0.0387
1.3	0.0169	0.0171	0.0266
2.0	0.0054	0.0050	0.0244

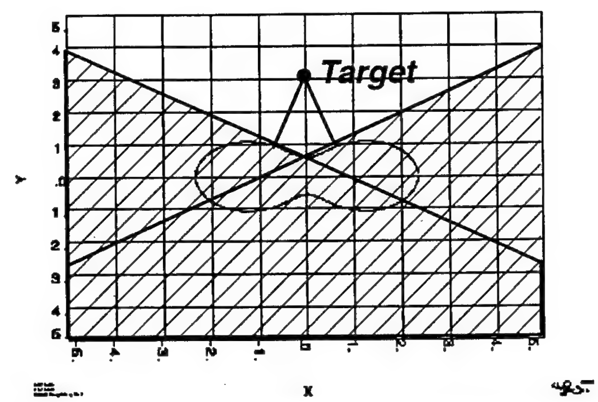


Figure 4. Polyhedral Approximation to Failure Region.

2.2 Discussion of the Estimation Algorithms

Realism could be added to the example of the preceding section by correctly modeling the propagation of the wavefronts and by adding probabilistic dimensions. For example, the following could be treated as random variables:

- The critical pressure ρ sustainable by the target.
- The spatial separation of the detonations.
- The angular orientation about the midpoint of the detonations.
- The magnitudes of the detonations.
- The time between detonations.

Both direct numerical integration and first order reliability approaches become less feasible as the dimension of the problem increases. Both suffer from the necessity to clearly specify failure regions; the latter requires additional effort to define bounding polyhedra which are themselves not easily integrated.

By contrast, the performance of Monte Carlo integration does not intrinsically depend on problem dimension. The coefficient of variation of the estimator (a measure of convergence) is $[(1-\theta)/\theta n]^{1/2}$, depending only on the probability being estimated, θ; and the number of iterations, n. While Monte Carlo is more robust than the other two approaches, in this respect there are issues which must be addressed:

Convergence

The number of iterations required to estimate a small probability is very large. The number of iterations required to maintain a given coefficient of variation, R, is approximately inversely proportional to the probability being estimated:

$$n = (1-\theta) / \theta R^2$$

$$n \cong 1/\theta R^2 \text{ for small values of } \theta$$

(Confident estimation to within a factor of 2 of the true value of a probability on the order of 1/10,000 requires roughly 10,000 trials.)

Fortunately, it is often possible to modify Monte Carlo by effective acceleration techniques, significantly improving convergence rates (see Reference 3 for discussions of stratified sampling, importance sampling and Latin hypercube sampling).

Cost

Even the fastest Monte Carlo algorithm may be prohibitive in applications where each iteration is computation intensive (requiring, for example, refined finite element analysis). Given sufficient continuity (or smoothness), a system can be approximated by a response surface constructed to interpolate analyses run at a selected grid of parameter points. With a response surface set, millions (even hundreds of millions) of Monte Carlo trials can be run at little additional cost. It may be countered that too many points are required to fit a high dimension response surface, but lacking the requisite system analysis, meaningful probabilistic estimates are simply not possible.

(It is emphasized that the above definition of "response surface" allows for more than a single linear or quadratic functional approximation.)

Numerical Traps

Good pseudorandom number generators are required for Monte Carlo.

While this seems a simple requirement, research has shown that common, older algorithms can generate sequences which cycle with very short periods. For example, an algorithm provided in Reference 4 based on the Multiplicative Linear Congruential (MLC) Method (Reference 5) can produce sequences with periods as short as 8,192 (Reference 6). Newer algorithms have been developed based on the Lagged Fibonacci (LF) Method with periods exceeding 10^{40} (Reference 7).

Also of concern are correlations between dimensions when points are generated by pulling $x_1(i)$ from the generator, then $x_2(i)$, and so on. Correlations invalidate the assumption of independence. Four examples of correlated sequences generated by the referenced MLC algorithm are shown in Figure 5. The LF algorithms seem not to suffer this problem.

Much effort has been spent developing algorithms derived from the first order and related second order reliability approximations and three commercially available computer pro-

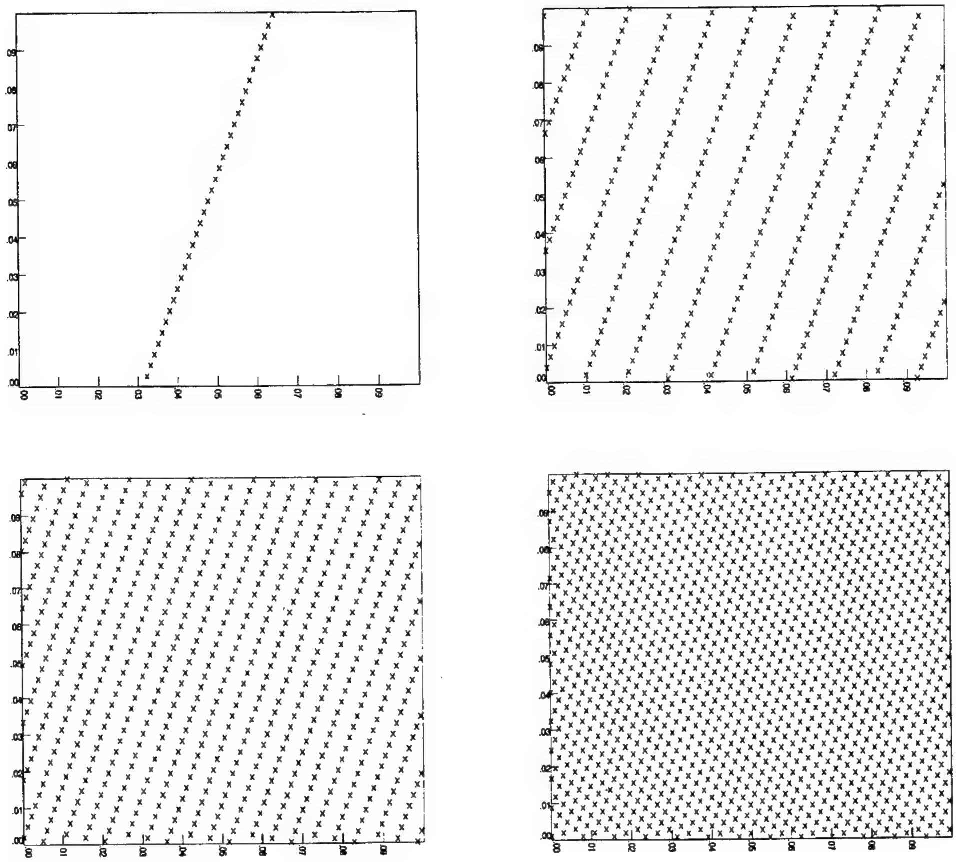


Figure 5. Correlated (x,y) Sequences Generated by MLC Algorithm.

grams are reviewed in Reference 8: NESSUS, PROBAN and CALREL (References 9–11). The comment is often made that there is no alternative when problems involve hundreds of random variables.

Approximation techniques can be effective. It is easy to calculate the probability, P, of a hypercube whose sides are independent random variables having known distributions. If it can be shown that the hypercube avoids the failure region, then 1–P upper bounds the failure probability. When this is an acceptable estimate, why do more?

Approximation techniques are not always good. First and second order reliability algorithms can be applied blindly since it is easy to assume a failure region to be bounded by a hyperplane even if it is not. Given the Section 2.1 example demonstrating that even simple phenomena can yield complex failure regions, caution should be exercised when computing a problem involving hundreds of random variables, even if there are no alternatives.

2.3 Probabilistic Design Analysis System (PDAS)

The algorithms reviewed in Sections 2.1 and 2.2 could be presented briefly given that the example involved a simple algebraic response function. A probabilistic analysis of a complex structure works conceptually the same. Controlling parameters are identified. These are varied and the response observed. A failure region is identified and its probability volume estimated.

Simplicity vanishes, however. All probabilistic algorithms require multiple analyses, and each analysis of a complex structure tends to be, for lack of a better word, complex. A typical turbine engine disk analysis requires geometric definition, finite element meshing, thermal and stress solutions, and life calculations. This must be preceded by a cycle definition: rotor speeds, gas temperatures, pressures and flow rates, requiring additional analysis. The analyses all require data: measured engine parameters, heat transfer coefficients, coefficients of expansion, tensile curves, fatigue curves, and crack growth rate curves.

PDAS was developed to:

- Manage multiple executions of the user's choice of analysis process, facilitating its application to design perturbations as needed to map a structure's response.
- Provide a range of integration approaches for estimating the structure's failure probability given its response to statistically distributed controlling parameters.
- Enable the structure's design to be optimized subject to a constraint on failure probability.

The mix of required analysis tools can vary and it is not currently practical to hard wire all mixes into a single computer code. Accepting the need for flexibility, PDAS was designed to be more like a language than a program – it does not define an analysis but provides words and concepts so that the user can define the analysis. There are core GEAE–developed modules for selected functions such as spreadsheet management, distributional handling, finite element model interfacing, and life calculations. There are also links to external functions such as Unigraphics, Patran, and ANSYS, ensuring the best tools can be applied to a given problem.

PDAS will be described in detail in Section 6. Appendix C will provide examples of three PDAS programs used for the calculations summarized in Section 2.1.

3.0 Probabilistic Fracture Mechanics

Traditional fatigue life approaches focus on limited numbers of critical component locations, estimating minimum lives at those locations based on calculated temperatures and stresses, lower bounds on material fatigue capability based on laboratory testing of carefully prepared simple geometry specimens, and a mechanistic model to relate the generally multiaxial, complex–cycle stresses of the part to the generally uniaxial, simple–cycle stresses of the specimens. Damage tolerance approaches (e.g. ENSIP, Reference 12) depart from this theme only in that they set criteria based on residual life from cracks assumed placed at the same component locations.

The methods which are used work to yield the best possible accuracy at the focused critical locations: 3–D models are used to calibrate stress concentration factors. Mission simulation programs generate local temperature/stress histories that may contain many thousands of partial cycles. K_t corrected missions enter into life algorithms where they are integrated with material data to yield a predicted number of cycles to failure.

The introduction of powder metal (PM) alloys into disk applications led GEAE, Pratt & Whitney, SNECMA and others to question the validity of traditional lifing methodology. Early PM alloys (René 95, Inconel 100, N'18) were recognized to be life–limited by inherent process–related inclusions. Unless suppressed by a surface treatment such as shot peening, a 50 sq mil inclusion falling on the surface of a part at a high stress location significantly impacts life. Efforts have been made through the years to improve the powder process, and more recent alloy development programs have produced materials such as René 88 DT which are more tolerant of inclusions. Even so, inclusions must remain a concern.

When inclusions play a dominant role, a statistical size effect is clearly implied. Large volumes are more likely to hold limiting inclusions than small volumes. Given this observation, GEAE resolved to pursue a new lifing methodology with a more solidly probabilistic basis. The Probabilistic Fracture Mechanics (PFM) program MISSYDD (MISsion SYnthesis given Defect Distribution) was begun and is now in its sixth generation. MISSYDD serves as a foundation for the PDAS development.

3.1 Outline of the PFM Risk Calculation

While many approaches can be considered for calculating a component's failure probability from its inherent inclusions, the following three step division of the procedure has advantages:

1. Calculation of the probability of failure by given a single inclusion of a fixed size occurring randomly in the component. This probability is geometric in that it involves the spatial distribution of stress and temperature. In principle, each point of the model has a well defined life dependent on the precise mission conditions, geometric constraints and material properties in the vicinity of the point. Clearly, none of these factors are firmly fixed by design. The mission may be variable so that stresses and temperatures fluctuate randomly. Free surfaces may vary within drawing tolerances. Material properties influencing life may differ from point to point and part to part. These factors must be recognized and, ideally, should be incorporated into the analysis. Given random placement of the inclusion, the probability of failure by a life N is the volume of material having life less than or equal to this N divided by the total volume.

2. Calculation of the probability of failure given a single inclusion from a distribution of sizes. The geometric failure distribution is integrated with the relative inclusion size distribution.

3. Calculation of the probability of failure given more than one inclusion. This is like flipping a coin:

   The probability of one head is 0.5. The probability of two heads is $(0.5)^2$. The probability of n heads is $(0.5)^n$, a very small number for n large.

   A component may survive a single inclusion with high probability, say 0.999999 (since the probability that the inclusion is large and in the wrong place is small). But there are many inclusions in powder alloy parts, all competing for failure. The survival probability given 100 inclusions is $(0.999999)^{100} \cong 0.9999$.

   If F is the failure probability given a single inclusion, then (1–F) is the corresponding survival probability. The survival probability given n independently competing inclusions is $(1-F)^n$; the failure probability is $1-(1-F)^n$. This simple expression is integrated with the probability that there are n inclusions to yield net failure probability. This last step depends on the average rate of occurrence (number per cubic inch) of inclusions in a part times the part volume.

Summarizing the risk equations

Let G(N, a) denote the geometric failure probability, the probability of failure by life N given a single inclusion of size a. Let s(a) denote the probability density function of the inclusion size distribution. G(N, a) and s(a) are integrated to yield R(N), the probability of failure given a single inclusion, randomly sized.

$$R(N) = \int_0^{\infty} G(N,a)\ s(a)\ da$$

To calculate the failure probability given competing inclusions, F(N); the Poisson model for inclusion occurrence will be assumed:
The probability that there are n inclusions in a component volume is $e^{-V\lambda}\ (V\lambda)^n/n!$ where λ the average inclusion frequency (number per cubic inch) and V is the volume.

$$\begin{aligned}
F(N) = {} & \text{Prob (1 defect present – it fails by N)} \\
& + \text{Prob(2 defects present – one or both fail by N)} \\
& + \text{Prob(3 defects present – one or more fails by N)} \\
& + \ldots \\
= {} & \frac{e^{-\lambda V}(\lambda V)^1}{1!}(1 - (1-R(N))^1) \\
& + \frac{e^{-\lambda V}(\lambda V)^2}{2!}(1 - (1-R(N))^2) \\
& + \frac{e^{-\lambda V}(\lambda V)^3}{3!}(1 - (1-R(N))^3) \\
& + \ldots \\
= {} & e^{-\lambda V}(e^{\lambda V} - 1 - e^{\lambda V(1-R(N))} + 1)
\end{aligned}$$

$$F(N) = 1 - e^{-\lambda V(R(N))}$$

3.2 Note on the Statistical Size Effect

The size effect concept has broader application. All materials initiate failure at limiting microstructural features (grains, carbides, nitrides). The confluence of microstructure and stress is to some extent statistical. Disks with 80 bolt holes will have a lower minimum life than specimens with single bolt holes (assuming the two geometries can be equivalently stressed).

Detailed studies of the size effect in wrought Inconel 718 have concluded that it is overwhelmed by non–statistical stratification of data between parts and between part locations (References 1 and 13). Statistical size effect is inconsequential if life distributions are narrow within the volumes of the stress–concentrated regions, but powder alloy life distributions can be very broad given the distributions of inclusion sizes, locations and behaviors.

4.0 Dealing with Variability: The Probabilistics Target

Figure 6 shows the comparisons among PFM–predicted and observed failure distributions for five specimen sets: cylindrical and hourglass, unseeded baseline, large and small seeded ... all peened and all tested at the same conditions of strain and temperature. The comparisons demonstrate the capability of the PFM model over the range of probabilities 0.01 to 0.99 when provided the right inputs of inclusion distribution and behavior. It is easily accepted that were the rate of inclusion occurrence orders of magnitude lower than the selected seeding densities (more like the real world), the failure probabilities would be proportionately lower in the absence of other failure mechanisms.

Significant efforts are being made to ensure the right inputs for design application of PFM. For example, Heavy Liquid Separation (HLS) analysis has been developed to quantify and control inclusion content. This work was presented at the 1994 Toronto meeting of the Metal Powder Industries Federation and published in the proceedings with the vision that HLS (or something like it) will become an industry standard (14). Also, an intensive seeded fatigue program providing a model for crack initiation at inclusions is ongoing. Some of the details of this work will be presented at the September 1996 Seven Springs Superalloys Conference.

Accepting that the technology is available and sufficiently mature to be applied to hardware, two challenges remain: an appropriate risk design level must be accepted and surprises must be avoided.

Conventional deterministic design methods yield products having nonzero risk of failure, but the risks are hidden. Given a design based on minimum properties (-3σ), it is seldom

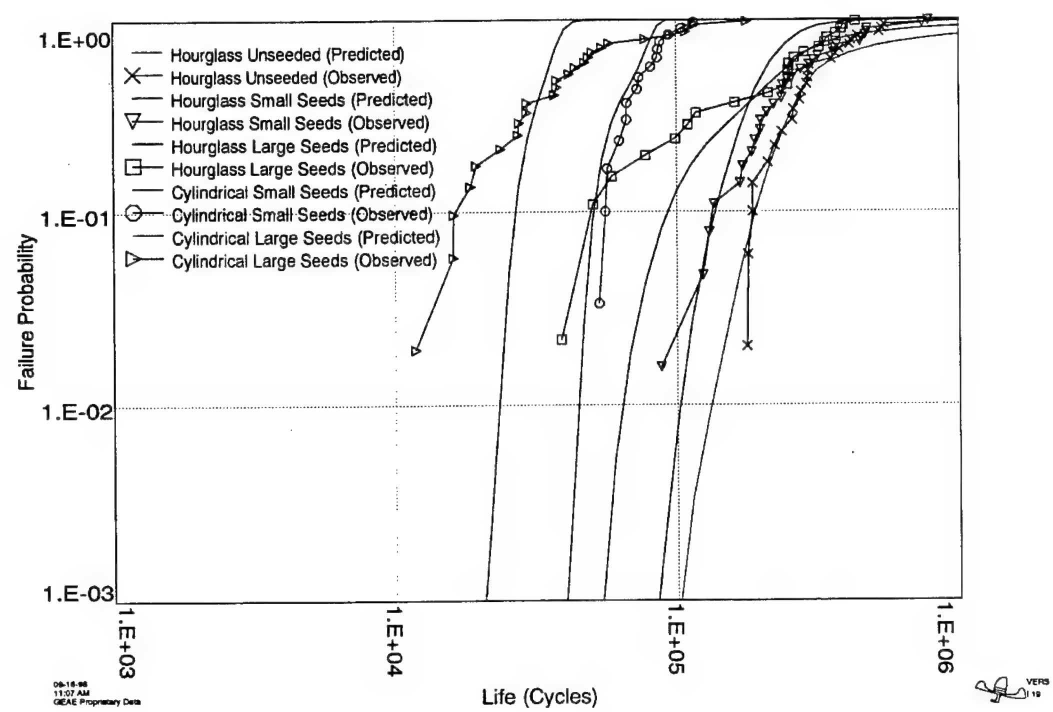


Figure 6. Peened Specimens Tested at One Temperature/Strain Condition.

noted that failure might occur 1.35 times out of 1000. In practice, conservatisms are usually compounded and safety factors applied, but there is always some risk, and there can be failures. By contrast, PFM quantifies risk; risk becomes a characteristic to be controlled and kept below an accepted level. In Reference 1, FAA statistics are used to support 1/1,000 or 1/10,000 probabilities at full life as appropriate for fatigue failure.

Given the right inputs, the seeded validation testing supports the capability of PFM to correctly predict risk. The second challenge addresses our ability to define the right inputs. Each failure distribution in Figure 7 corresponds to a perturbation in part geometry (the nine perturbations overlaid as shown). Designing to a low failure probability on the nominal curve ignores the fact that it may be the wrong curve for any given part. Many other factors can move PFM calculated failure curves and create similar concerns.

An aircraft engine component design analysis is based on many knowns: basic geometry is well defined as is the envelope of speeds and gas temperatures defining the cycle for a given application. Metal stresses and temperatures can be calculated with reasonable precision using finite element and finite difference methods. For PFM calculations, the average inclusion distribution can be determined, as can the average incubation behavior of the inclusions (cycles to crack initiation) and the average crack growth properties (growth from initiation to failure).

But the example of Figure 7 illustrates the need to recognize sensitivities to deviations from nominal values of design parameters. The example (though admittedly extreme) demonstrates the potential impact of manufacturing tolerances. The effect of variations in other parameters may be as significant or more so: Actual engine usage will vary from flight to flight, from base to base, or from pilot to pilot.

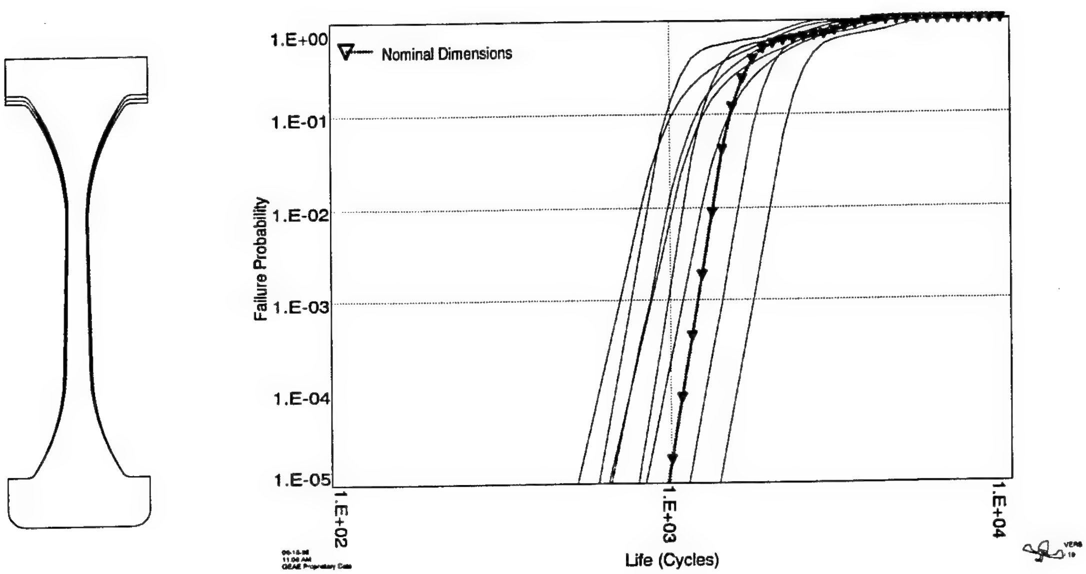


Figure 7. Predicted Failure Distributions for Nine Disk Perturbations.

Stresses and temperatures may be reasonably calculated, but not exactly (temperatures within ±25°F at steady state conditions, within ±50°F for transients; critical material properties can vary significantly over 50°). There is variability in inclusion distribution and in inclusion behavior.

There will also be surprises: deviations from manufacturing tolerances, unexpected applications, analysis oversights, methodology shortfalls, and material anomalies.

The target in Figure 8 suggests a strategy:

1. The Bull's–eye – Apply what we know as well as we know how to. PFM does provide a measure of product capability given nominal assumptions.
2. The Middle Ring – Evaluate sensitivities to known deviations and bound the results. Assume the left–most curve in Figure 7.
3. The Outer Ring – Back off from the calculations until accumulated experience justifies doing otherwise. Apply a safety factor on computed minimum life.

The relative proportions of the target circles can vary as a fleet matures and as probabilistic methodology improves. Middle ring sensitivities may be pushed into the bull's–eye if the variability can be statistically defined and properly incorporated. The outer ring may shrink given experience and/or implementation of production controls aimed at preventing surprises.

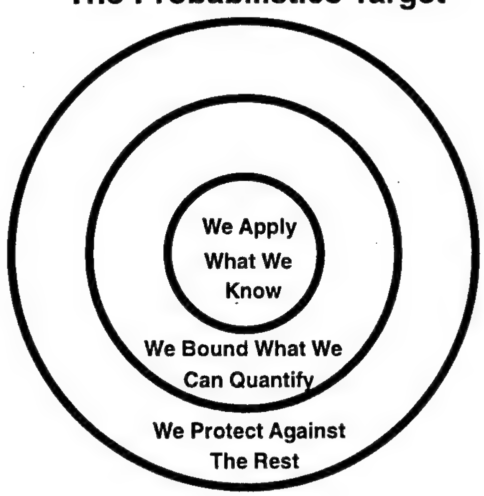
4.1 Example – Factoring In Dimensional Variability

The risk curves in Figure 7 define a response surface dependent on part dimensions X_1, X_2 and X_3. Assume that manufacturing deviations in these parameters can be modeled as random variables with joint density $\rho(x_1, x_2, x_3) = u(x_1)\, v(x_2)\, w(x_3)$ (Figure 9). Let $F(N: x_1, x_2, x_3)$ denote the conditional probability of failure given geometry defined by x_1, x_2 and x_3, and integrate against the joint density:

$$\int F(N : x_1,\ x_2,\ x_3)\rho(x_1,\ x_2,\ x_3)dx_1,\ dx_2,\ dx_3$$

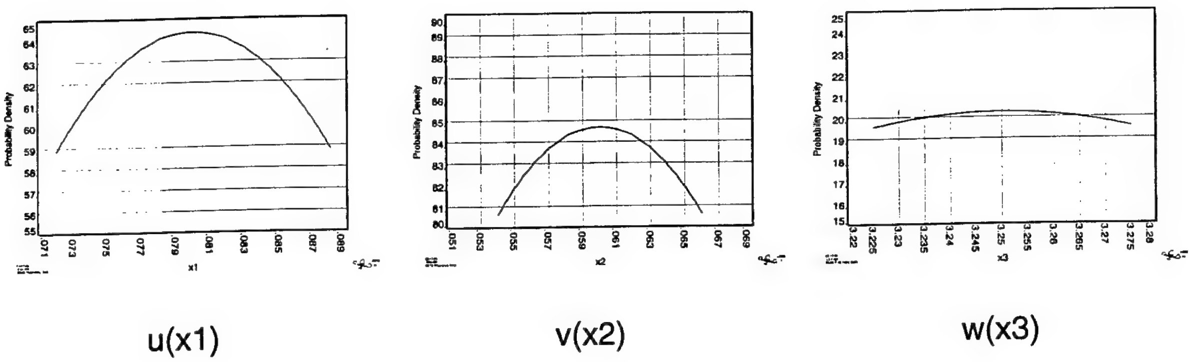
This yields an average distribution, the Integrated Result indicated in Figure 10. Designing to this distribution captures the variability, but removes the arbitrary conservatism entailed in designing to the minimum (left–most) curve.

The Probabilistics Target



Known	*Sensitivities*	*The Rest*
Basic Geometry	Manufacturing Tolerances	Uncaptured Deviations
Typical Usage	Mission Envelope	Unexpected Application
Representative Temperatures	Delta Temperatures	Analysis Oversights Methodology Shortfalls
Representative Stresses	Delta Stresses	Boundary Condition Errors
Average Flaw Distribution	Lot–To–Lot Variability	Analysis Oversights Methodology Shortfalls
Average Incubation	Lot–To–Lot Variability	Uncaptured Deviations
Average Crack Growth	Lot–To–Lot Variability	

Figure 8. Strategy for Implementation of Probablistics.



u(x1)   v(x2)   w(x3)

Figure 9. Dimensional Distribution Probability Density Functions.

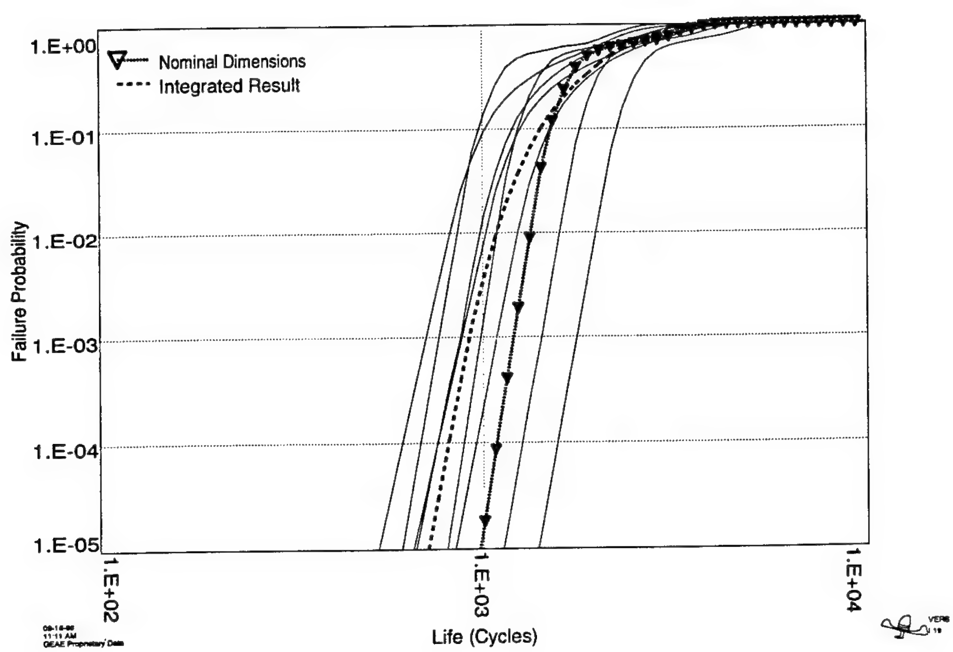


Figure 10. Risk Response Surface Integrated against Dimensional Distributions.

4.2 Example – Factoring in Variability in Inclusion Distribution

As was discussed in (Reference 14), Heavy Liquid Separation (HLS) is a PM cleanliness evaluation technique which has been developed by GE Aircraft Engines and Wyman–Gordon. A high density liquid is used to float ceramic inclusions from a PM sample for characterization by automated SEM/EDAX analysis. The number, size distribution, and chemistries of the recovered inclusions can be used for statistical process control (SPC) monitoring and improvement of PM lot cleanliness, and as a quality control screen to help ensure an acceptable level of cleanliness for PM used in critical applications.

Inclusion distributional models are core to the evolving risk analysis methodology. For any given material, MISSYDD assumes a homogeneous inclusion distribution (a single distribution which can be generically applied to all production lots). Sufficient data exists to demonstrate the inadequacy of this assumption.

HLS analysis of a large number of powder lots yielded considerable scatter between samples (Figure 11). A conservative (middle ring) application of this data would use an upper bound distribution for PFM calculations. Design life would be based on the acceptable risk level assuming the part was manufactured from the dirtiest powder likely to be encountered.

Conservatism can be reduced (movement towards the bull's–eye) if the observed scatter can be integrated into the risk calculation. Assume that lot variations can be modeled as a random vector $\mathbf{x}$ with joint density $\rho(\mathbf{x})$. Let $F(N: \mathbf{x})$ denote the conditional probability of failure given the inclusion distribution defined by $\mathbf{x}$, and integrate against the joint density:

$$\int F(N : \mathbf{x})\ \rho(\mathbf{x})\ d\mathbf{x}$$

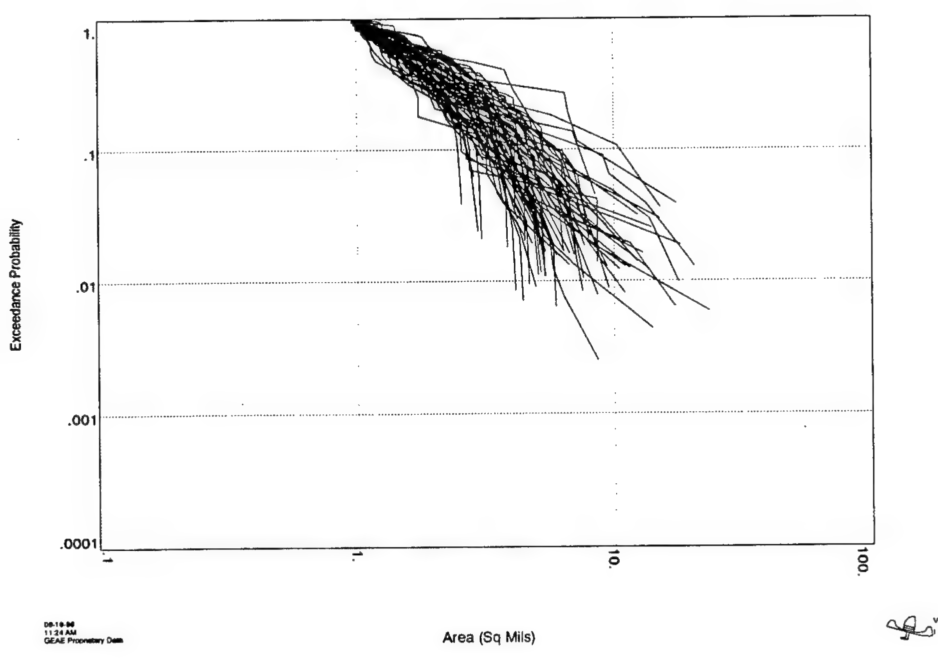


Figure 11. Lot–to–Lot Variability in Inclusion Distribution.

This would yield the desired refinement. Unfortunately, the parametrization is not available, much less the joint density $\rho(\mathbf{x})$. Fortunately, it can be shown that:

$$\int F(N : \mathbf{x})\ \rho(\mathbf{x})\ d\mathbf{x}$$

$$\leq 1-\exp\left[-\lambda V \int R(N : \mathbf{x})\ \rho(\mathbf{x})\ d\mathbf{x}\right]$$

$$= 1-\exp\left[-\lambda V \int G(N, a) \int s_{\mathbf{x}}(a)\ \rho(\mathbf{x})\ d\mathbf{x}\ da\right]$$

(Jensen's Inequality for the exponential function, Reference 15). The integral $\int \mathbf{x}(a)\rho(\mathbf{x})d\mathbf{x}$ is an average inclusion distribution estimated by combining all data into a single distribution having density s(a) (see Figure 11). The inequality is rewritten:

$$\int F(N : \mathbf{x})\ \rho(\mathbf{x})\ d\mathbf{x}$$

$$\leq 1 - \exp\left[-\lambda V \int G(N, a)\ s(a)\ da\right]$$

Thus, it is sometimes possible to factor in lot–to–lot scatter without actually quantifying it. Using the average distribution produces an estimate of failure probability which is conservative, but not as conservative as would result from assuming a worst–case distribution.

4.3 Example – Factoring in Variability in Inclusion Behavior

The PFM risk algorithm as presented in Section 3 assumes that the life of any given inclusion at any component location can be predicted exactly. In reality, there is scatter about the predicted life due to many factors including variability in inclusion shape, variability in matrix properties, and variability in actual loading.

Because this scatter has not been integrated into the calculations, the Figure 6 peened data predictions capture the average behavior of the observed distributions but not their breadth. The following model is a simple correction:

1. Assume that the scatter can be modelled by a distribution of life multipliers: Let μ denote the ratio of predicted to observed life for a single inclusion, and let $m(\mu)$ denote the probability density function of the μ distribution.
2. Assume also that the distribution of μ predominantly represents testing variability (i.e. specimen–to–specimen rather than inclusion–to–inclusion).

The distribution of life is derived from the basic PFM risk algorithm as a kind of convolution:

$$\int \left[1 - e^{-\lambda V \cdot R(N\mu)}\right] m(\mu)\ d\mu$$

Figure 12 shows that a Weibull distribution can be found for μ which fits the predicted/observed ratios of the cylindrical large and small seeded data sets of Figure 6. Convolution of this distribution with the calculations in Figure 6 yields better agreements between predictions and observations (see Figure 13). Figure 14 shows comparisons for other data sets. Note that the calculations for the unpeened data sets in Figure 14 appear to capture the competition between surface and subsurface initiated failures (surface initiated failures have shorter lives but occur with lower probability).

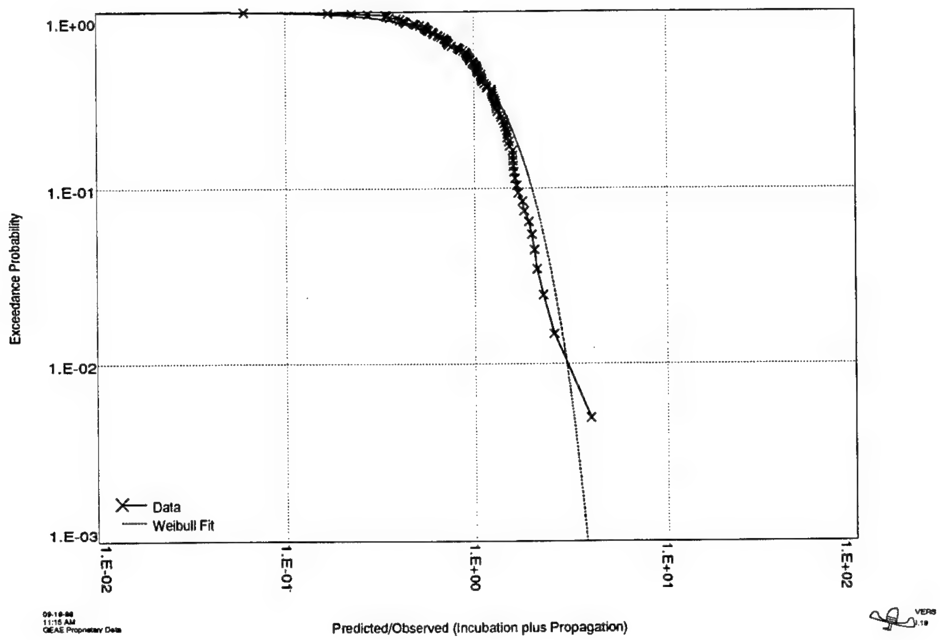


Figure 12. Weibull Fit to Predicted/Observed Distribution.

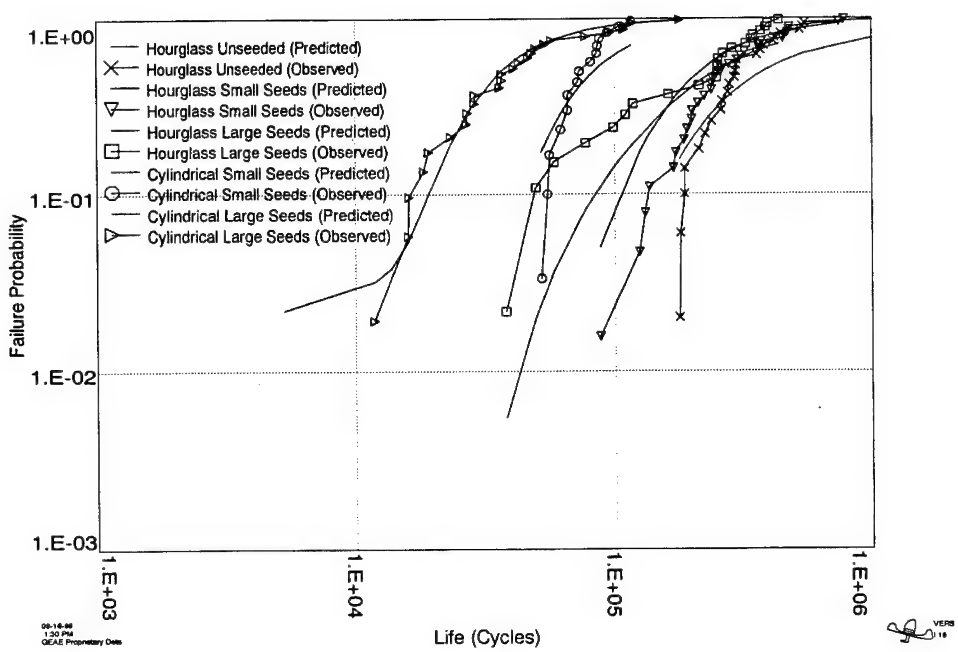


Figure 13. Material Variability Integrated into Specimen Predictions of Figure 6.

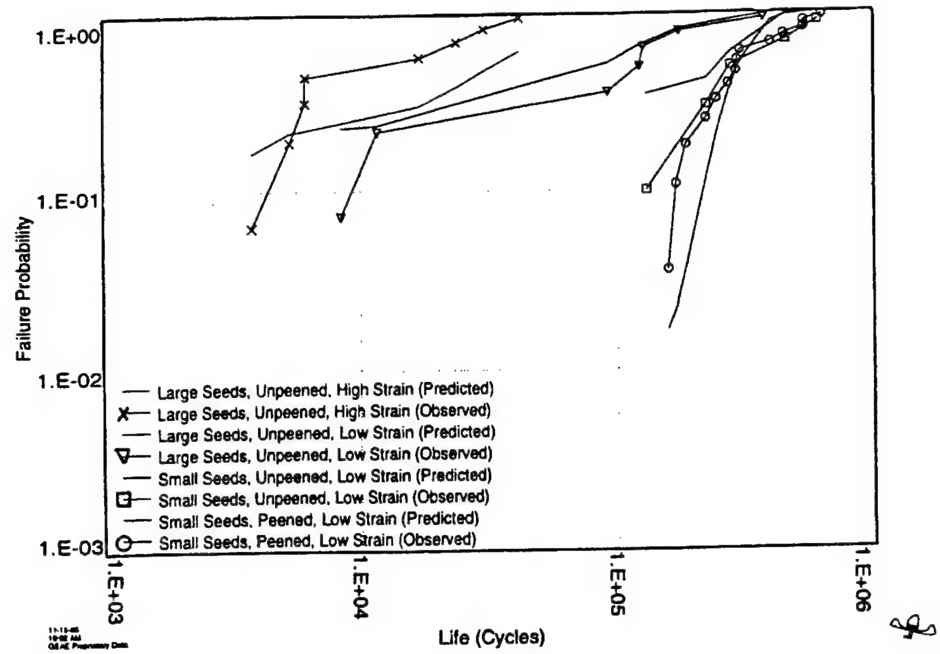


Figure 14. Material Variability Integrated into Other Cylindrical Specimen Data Sheets.

5.0 Statistical or Not?

Improvements as described in the last three examples can provide better analysis of middle–ring sensitivities, but only if the sensitivities can be properly modeled statistically. Probabilistics are often sought as cures for all variability, but they are strictly useful only in situations where randomness is well defined.

Probability and statistics deal with variables (measurable quantities) which are random for individual samples, but which are predictable (on the average) for larger samples – quantities for which the empirical distributions converge for large samples to fixed forms which may be parametric (binomial, normal, Weibull) or non-parametric. Statistical samples are generated under controlled conditions and analyzed to infer the underlying distribution; probabilistic theory is applied to predict the effects of the randomness under more complex circumstances.

Qualitatively, risk is a condition of uncertainty the potential result being undesirable. In attempting to quantify risk, probability theory is called on. This assumes that there is predictability even in randomness, which is sometimes true. We refer to physics and chemistry and the success of statistical and quantum mechanics, or to genetics and gambling where much simpler combinatoric predictions work on average. In these applications, populations may be sampled repeatedly and repeatably (the populations are open). Their statistical distributions can be derived from basic principles or inferred by preliminary sampling, and then applied to predict the results of subsequent sampling.

A population of aircraft engines is closed rather than open; a few thousand will be made over a span of several years. It is closed and probably not even stable. During production, design revisions may be made, melting and forging practices may be modified, subcomponent suppliers may change. Consequently, computed risks, while stated as probabilities, are usually not verifiable. A predicted 1/1,000 risk of failure can be disputed if enough fail, but can not be verified, and while 1/10,000 may be lower risk than 1/1,000, the difference can not be demonstrated. Decisions are qualitative despite the quantitative language.

Component integrity is dependent on many factors, some of which are at least approximately statistical (inclusion distribution), but many of which are inherently non–statistical. For example, a design modification in a derivative engine may be mistakenly assumed benign based on its ancestry, and no further analysis performed. In another example, a stress concentration may be underestimated based on an oversimplified textbook solution or on a poorly refined finite element model. While a probabilistic design framework can sometimes facilitate evaluation of sensitivities by quantifying risk as a function of the potential error, such errors can probably not be usefully accommodated by statistical analysis.

Safety factors may be applied to computed minimum component lives upon launch of new designs or design revisions as a hedge against analysis errors and oversights. The factors may be essentially arbitrary, or they may be tied to some qualitative judgement of analysis goodness or based on field inspection intervals. In any case they need not remain fixed. Analysis refinements and improvements in material understanding can be counted on, and while operating experience may uncover problems, it will also support a good design's fundamental integrity. In most cases, factors will increase with time.

Probabilistics work if properly applied. They can produce designs which are robust against

failure since risks and sensitivities are quantified. In some cases they may yield economies in weight or manufacturing requirements. In other cases weight may be added or manufacturing requirements stiffened to lower risk to an accepted appropriate level.

The probabilistic tools do work, but must be applied at a proper level. Some parameter sensitivities cannot be incorporated probabilistically (at least not at the outset) and must be bounded. Some allowance must also be made for oversights. Thus, conservatism should not be abandoned, but may be reduced in time given dedicated efforts. While it is prudent that caution be exercised in reducing conservatism, it is equally prudent to pare away unnecessary conservatism which wastes resources and impedes society's growth into the future.

5.1 Application Strategies

It is given that some components cannot be designed for the largest possible defect or the worst possible material properties, and risk based design provides a reasonable alternative. The examples of integrated risk analyses in Section 4 cover situations which could be different.

As demonstrated in Section 4, variability in part dimensions yields parts with a range of failure distributions, and while dimensional variability can only be controlled within limits, it can be quantified for all parts produced. Similarly, variability in inclusion distribution from powder lot to powder lot yields a range of failure distributions; it cannot be controlled, but can be quantified within the limits of sampling variability.

There is a difference between fielding 10,000 parts, expecting (based on an estimated failure probability of 1/10,000) that 1 will fail, and fielding 10,000 parts, expecting (based on an inspection) that 1 particular part will fail. The latter would clearly be unethical. The distinction is less clear if an inspection can quantify a parameter which might indicate higher risks for some parts and lower risks for others.

Consider the following alternative strategies:

- Retire parts at the 1/10,000 life for the absolute *worst* component where worst covers all parameters which can be measured.

Of course *worst* never really means *worst*. *Worst* means somewhere on the edge of a sharply defined process window. It is tacitly assumed that the measurements are exact and that a point inside the window is always better than a point outside. It is also assumed, tacitly, that parts are rejected if they fall outside the window.

When the assumptions are met, this middle ring solution of the probabilistic target has merit from the point of view of safety. It is wasteful, however, if there is a large difference between the worst part and one that is likely to be made.

- Retire parts at the 1/10,000 life predicted by a fully integrated risk analysis. This yields an average 1/10,000 failure probability, but individual parts will have risks higher or lower than 1/10,000.

This second strategy may be acceptable if the assumptions behind the first strategy are in question. For example, HLS is useful as a quality control tool. It can detect trends, and can detect a process gone out of control. Given samples from many powder lots, it also provides an average picture of the inclusion distribution which can be used for PFM calculations. However, HLS cannot be used to collect enough data for individual lots to confidently define a window; the measurement is not exact. (Sampling variability is demonstrated in Appendix B.)

This second strategy may be acceptable if it is judged that there is adequate conservatism built elsewhere into the analysis. For example, it was demonstrated in Section 4.2 that applying the full breadth of the average inclusion distribution overestimates failure probability.

This second strategy may be essential to managing a field problem. For example: A fillet radius is found to be a critical dimension only after several hundred engines have been fielded. A badly undersized radius would be expected to lead to cracking long before the first planned shop visit. Earlier inspections may be needed to assure that offending parts are weeded out for rework. Assuming that all fillet radii are undersized would call for immediate grounding of the fleet. Based on a measured distribution of fillet radii, an inspection interval may be set which holds the risk below 1/10,000.

- Retire parts at the 1/10,000 life predicted by a fully integrated risk analysis, but reject parts having failure probabilities exceeding 1/10,000 at this life.

This third strategy is less conservative than the first, but addresses a concern with the second strategy by holding risk at an acceptable level for all components. Given a robust design, very few parts will be rejected.

- Retire parts at individually calculated 1/10,000 life limits.

This strategy seems to be optimal. It holds risk at a uniform level for all components, and also utilizes full part life.

It will be argued that it is too complicated, but parts are already tracked by cycles and hours flown. Adding another number onto a spreadsheet is not a major burden and seems completely consistent with the USAF interest in life monitoring. Balancing engines so that teardowns are efficiently managed will be necessary, but the net cost savings must be tremendous if there are large part–to–part differences.

6.0 PDAS Template Programming

MISSYDD was developed to perform only very specific analysis functions. Other programs were needed for concurrent tasks of data analysis, algebraic manipulations and simulation studies. While commercially available packages were sometimes available, they were often found wanting, and special programming had to be developed.

Local programs are generally written in Fortran. Large portions of the programs are generally very similar from one to another. There may be blocks of input/output code, a sorting routine, random number generators, and so on. Much of the overlap can be eliminated by linking programs with libraries of standard subroutines; this requires careful attention to assure that calling sequences be exactly right. Arrays must be dimensioned, files opened and closed. Often the lines dedicated to actual computations are outnumbered many times by the wrapping.

Two developments at GEAE suggested an alternative:

1. The System for Integrated Engineering Structural and Thermal Analysis (SIESTA) development team refined the concept of keyword–defined input and output.
2. Greg Blanc developed an expression evaluator package which interprets Fortran–like algebraic expressions. Using these concepts (and many of the routines) as a starting point, a new programming language has been developed which greatly simplifies many analysis tasks. Simple commands replace blocks of code. For example:

```
table 1 input
x.dat
```

causes the file x.dat to be opened and the keyword–defined data to be entered into a table identified as 1.

```
table 1 set y = EXP(LOG(X) + SIN(X**2))
```

creates a y–column in table 1 by transforming the x–column by the expression following the equal sign.

```
distribution make Y (table) 1 x
```

produces an empirical distribution, labeled Y, from the x–column data in table 1.

```
x = (random) Y
```

assigns to parameter x a random value from the Y–distribution.

Other commands enable manipulation of a MISSYDD database. Models, missions, and computations can be defined and operated on as directed by simple template programs.

External functions are also significantly assisted by GEAE's SIESTA software. Component designs generally require many analysis steps for accurate deterministic predictions: parametric geometry definition, finite element thermal/stress/displacement analysis (both elastic and plastic (up to burst)), fatigue, and fracture mechanics calculations. The best deterministic tools are required to maximize accuracy in a probabilistic analysis.

SIESTA provides unified access to a range of disciplines. For example, ANSYS may be chosen for geometry definition and thermal/stress/displacement analysis, or it may be decided to call Unigraph-

ics and Patran for modeling and meshing followed by ANSYS for the solution. This flexibility allows concentration on the leading edge of development needs.

The following is a brief summary of the template language.

6.1 Keywords

Keywords control program flow. They can be entered in any case, and only the first four characters are significant (`TABLE`, `table`, `Table`, `TABL`, `tabl`, `Tabl` are all equivalent). The following are the supported primary keys:

TABL	Table Manipulation
EXPR	Expression Definition
ALGE	Algebraic Operations
DIST	Distribution Handling
REGR	Regression Functions
LOOP	Loops
OPTI	Optimization
PLOT	Plotting
MODL	Model Library
MSSN	Mission Library
COMP	Computation Library
LIFE	Life Calculations
RISK	Risk Calculations

Entering a primary keyword changes control to its menu of secondary keywords. Secondary keywords can follow on the same line or on following lines. For example:

```
table 1 input
x.dat
table 1 set y = EXP(LOG(X) + SIN(X**2))
```

is equivalent to

```
table 1
input
x.dat
set y = EXP(LOG(X) + SIN(X**2))
```

A few secondary keywords lead to menus of tertiary keywords.

6.2 Table Manipulation

A table is an array of numbers. Each column is a parameter. The rows are individual values. Example:

Row No.	X	Y	Z
1	2	1	21
2	1	2	12
3	2	2	22
4	2	1	21
5	1	1	11
6	2	1	21
7	1	1	11
8	1	2	12
9	2	2	22

Tables are identified by integer labels from 1 to 100 specified by the user after the primary keyword. For example:

```
table 10
```

Subsequent commands will look for direction from the `TABLE` menu and will address table 10 until the next occurrence of a primary key.

Tables may be referenced to define digitized functions or parameter distributions, and as data sets for regression analysis or plotting functions.

Tables may be defined by ASCII files or built by the template program. Columns may be defined or redefined by functional transformations. For example:

```
set U = exp(X + Y + Z)
```

Column `U` is defined by the expression to the right of the equal sign where `X`, `Y` and `Z` are constants or other table columns.

Rows may be flagged on or off for subsequent operations based on general lists of conditions. For example:

```
flag X .gt. 100 X .le. Y .le. Z
```

where `X`, `Y` and `Z` are constants or table columns. Rows satisfying the multiple inequalities are turned on.

Tables may be sorted or grouped by column values. Tables (or specified table columns) may be output to ASCII files.

6.3 Expression Definition

Expression evaluation routines have been incorporated into PDAS, enabling the application of algebraic functions for a variety of purposes. Digitized functions can also be defined with up to three independent variables.

Expressions play a part in parameter definitions, as integrands, as constraints and as regression models. Incubation in the life calculation module is also defined by expressions in terms of stress, temperature, inclusion area and inclusion depth. In the future it is planned to enable user definition of other material properties by expressions (e.g. crack growth rate curves, probability of detection curves, inclusion distributions).

The following is the basic format used for expression definition:

`EXPR {expression ID}:Expression`

`EXPR` is the primary keyword which initiates definition. The *ID* is an optional four character case–insensitive label. The colon points to the beginning of the expression. If the *ID* is included, the expression is stored in an expression table initialized at the start of template execution (up to 100 expressions can be so entered). If the *ID* is omitted, the expression is processed and held as the current expression for subsequent operations.

Expression is either an algebraic function (in capitals) or a digitized function defined by a previously input file. The rules for definition follow.

6.3.1 Algebraic Functions

Expression has the form of a Fortran expression using the operator set {+, –, *, / and **} and the function set {`sin, asin, cos, acos, tan, atan, cosh, sinh, tanh, log, log10, exp, sqrt, abs, unsf`} (`unsf` is the unit step function: `unsf(x) = 0` for $x \leq 0$, `unsf(x) = 1` for $x > 0$).

Parentheses within an expression are used to determine precedence of operation. The parentheses in the expression must balance – there must be as many right parentheses as left parentheses.

The equality symbol = is used to define relations to be fit in regression analyses.

6.3.2 Digitized Functions

Expression has the form:

`(DIG) (TABLE) {ID} {dep. var} [format] {ind. var 1} [format] ...`

Up to three independent variables can be considered.

[*format*] is optional. The following are recognized: `(LIN)`, `(LOG)`, `(POW)` and `(POL)`. The default is `(LIN)`.

`(LIN)` treats the variable as linear in the interpolation, `(LOG)` as logarithmic.

`(POL)` treats the variable as an angle in radians for a polar interpolation.

`(PWR)` does a power function interpolation (i.e. `dep = a*ind**b`).

Only `(LIN)` and `(LOG)` are applicable to the dependent variable.

The interpolation begins with the last independent variable. Independent variables are assumed to be layered in the following sense: Each combination of the first *p* variables must be adequately represented in the *p+1* variable to enable interpolation.

Example:

```
table 1 input
```

inputs the following data file:

RAVG	RMIN	THETA	TEMP
169.4	165.5	0	1200
94.7	75.7	0.24498	1200
55.8	44.6	0.59031	1200
48.1	38.5	0.7854	1200
44.3	35.4	1.5708	1200
109	103.6	0	1400
87.2	69.8	0.24498	1400
57.3	45.9	0.59031	1400
50.9	40.7	0.7854	1400
45.1	36.1	1.5708	1400
77.9	70.3	0	1600
76.1	60.9	0.24498	1600
54.3	43.4	0.59031	1600
46.7	37.3	0.7854	1600
43.4	34.7	1.5708	1600
41	33.8	0	1800
31.8	25.4	0.24498	1800
27.9	22.3	0.59031	1800
26.9	21.5	0.7854	1800
33.8	27.1	1.5708	1800
16.6	13.8	0	2000
15.6	12.5	0.24498	2000
15.3	12.2	0.59031	2000
14.9	11.9	0.7854	2000
19.1	15.3	1.5708	2000
12.2	10.5	0	2050
11.5	9.2	0.24498	2050
10.8	8.9	0.59031	2050
11.3	9	0.7854	2050
14.4	11.5	1.5708	2050
8.9	7.2	0	2100
7.8	6.3	0.24498	2100
7.4	5.9	0.59031	2100
7.4	5.9	0.7854	2100
11	8.8	1.5708	2100
4.1	3	0	2200
3.3	2.6	0.24498	2200
3.1	2.4	0.59031	2200
3	2.4	0.7854	2200
4.8	3.8	1.5708	2200

```
EXPR EX1 : (DIG) (TABLE) 1 RMIN TH (POLAR) TEMP
```

When evaluated, new values of RMIN are calculated for each TH by linearly interpolating with respect to TEMP. The new values of RMIN are then polar interpolated with respect to TH.

6.3.3 Equalities

Equalities have the following forms:

A = *value*	Initialization.
A = (EXP) {*expression ID*}	Expression evaluation.
A = *Expression*	Expression evaluation.

```
Examples:
A = 1
expr A : X + Y + Z
A = (exp) A
A = X + Y + Z
A = (dig) (table) 1 Y X (pol)
```

A = (MIN) *list*	List minimum
A = (MAX) *list*	List maximum

```
Examples:
A = (min) X Y Z
A = (max) A 1
```

A = (TABLE) *ID* (MIN) *column*	Column minimum
A = (TABLE) *ID* (MAX) *column*	Column maximum
A = (TABLE) *ID* (SUM) *column*	Column sum

Examples:

X	Y	Z
2	1	21
1	2	12
2	2	22
2	1	21
1	1	11
2	1	21
1	1	11
1	2	12
2	2	22

```
A = (table) 1 (min) X
yields: 1
A = (table) 1 (max) Y
yields: 2
A = (table) 1 (sum) Z
yields: 153
```

A = (DER) {*expression ID*} WRT *list*	Expression derivative with respect to the list variables.
A = (INT) {*expression ID*} WRT *list*	Expression integral with respect to the list variables.
A = (RAN) *parameter*	Random value from a previously entered parameter distribution.
A = (RAN)(EXP) {*expression ID*}	Random expression evaluation, some or all of the expression parameters being random variables.
A = (MODL) *parameter*	Model data such as volumes or surface areas, stress components, or temperatures at specific locations or averaged over model sections.
A = (LIF)	Life calculation at specific model locations.
A = (QUA) *quantile*	Survival or failure probability quantile following risk calculations

6.4 Algebraic Operations

Expressions may be rearranged symbolically.

```
| Examples:
|
| expr aaaa : X*(X*(X*(X*(X+1)+1)+1)+1)+1
| algebra prep aaaa
| algebra write aaaa
|
| produces:
|
| X*X*X*X*X+X*X*X*X*1+X*X*X*1+X*X*1+X*1+1
|
| expr bbbb : exp((sin(A) + B + C)*(sin(A) + B + C))
| algebra prep bbbb
| algebra write bbbb
|
| produces:
|
| exp((sin(A)*sin(A)+sin(A)*B+sin(A)*C+B*sin(A)+B*B+B*C &
| +C*sin(A)+C*B+C*C))
```

(This operation is performed internally prior to an integration of an algebraic function.)

Expression variables may be replaced with current parameter values.

```
| Example:
|
| expr cccc : A*(B + SIN(C + D))
|
| B = 11
| D = 22
|
| algebra replace B D
| algebra write cccc
```

```
|
| produces:
|
| A*(11.0000000+SIN(C+22.0000000))
```

Limited matrix operations may be performed (currently, only symmetric matrix diagonalization). Let X be the matrix key an n be the matrix order. The matrix is defined by the n^2 parameters:

```
X11                 ...   X1n
X21   X22
X31   X32   X33
Xn1   Xn2   Xn3   ...   Xnn
```

The current values of the super–diagonal elements determine the symmetric matrix to be diagonalized. The eigenvalues determined by the operation are placed in the parameters EIG1, EIG2, ..., EIGn. The eigenvectors are placed in the column parameters X1i, X2i, X3i, ..., Xni (i from 1 to n).

6.5 Distribution Handling

A good many PDAS operations involve statistical distributions of one type or another, inclusion distributions feed into risk calculations. Functions are integrated against various parameter distributions.

Distributions are stored in one of three digitized formats (X is a random variable):

1. Cumulative Format

As ordered pairs (x, C(x)) where C(x) = Probability($X \leq x$).

2. Exceedance Format

As ordered pairs (x, E(x)) where E(x) = Probability($X \geq x$).

3. Density Format

As ordered pairs (x, r(x)) where r(x) = dC(x)/dx = –dE(x)/dx.

$$C(x) = \int_{-\infty}^{x} r(u)\ du$$

$$E(x) = \int_{x}^{\infty} r(u)\ du$$

There are also specialized statistical models: The Poisson or Dirtiness models for material inclusion distribution are examples (Reference 14).

6.6 Regression Functions

Parametric models defined by Fortran–like equations can be least squares fit to data A model is referenced by its expression ID and the data held in a table. The regression can include all active rows

of the table or it can proceed iteratively over groups of rows. The resulting parameter estimates can be written to a table.

```
Example: (Reference 16.)

The data was generated from the equation:
y = 1.0 - x + 0.2*(x**2) and random errors added:
```

X	Y
0.050	0.956
0.110	0.890
0.150	0.832
0.310	0.717
0.460	0.571
0.520	0.539
0.700	0.378
0.740	0.370
0.820	0.306
0.980	0.242
1.170	0.104

```
expr aaaa : y = a + b*x + c*(x**2)

regr fit (expr) aaaa to 1

yields:

A : .9979684E+00
B : -.1018042E+01
C : .2246821E+00
```

6.7 Loops

One of the most powerful features of any programming language is the loop, the ability to repeat the same sequence of operations a specified number of times. Loops (and nested loops) can be incorporated into PDAS templates in a number of ways:

Programs can loop over a numerical range.

```
Example:
loop from 1 to 10 by 0.5
Executes the loop for the following values of a counter:
1, 1.5, 2, 2.5, ..., 10.
```

Programs can loop over the active rows of a table.

```
| Example:
| loop over (table) 1
| Executes the loop once for each active row of table 1.
```

Programs can loop until a general list of conditions is met.

```
| Example:
| loop until U .le. 2 (expr) A .ge. (expr) B
| Executes the loop until the parameter U is less than or
  equal to 2 and expression A is greater than or equal to
  expression B.
```

Programs can loop while a general list of conditions is met.

```
| Example:
| loop if U .le. 2 (expr) A .ge. (expr) B
```

Executes the loop if the parameter `U` is less than or equal to 2 and expression `A` is greater than or equal to expression `B`.

While template loops are useful, it must be emphasized that they are not efficiently compiled. PDAS cycles through loop blocks interpreting and executing the instructions one by one. While working analyses have been made with millions of loop iterations, they can take days to run. PDAS template programming is a good tool for many simple everyday applications, for some complex once–only analyses and for general development prototyping. It can not compete with dedicated programs targeting single tasks.

6.8 Optimization

A flexible optimization module was developed based on commercially available ADS and IMSL algorithms (References 17–19). Definitions of constraints and bounds can be made with greater freedom than allowed by the conventions of either commercial package.

Secondary Keys:

`INIT` Initializes an optimization.

`ALGO` Specifies the algorithm: `ADS` (the default) or `IMSL`. The ADS selection allows optional specification of strategy, optimizer, one dimensional search and print options as controlled by the respective secondary modifiers `ISTRAT`, `IOPT`, `IONED` and `IPRINT` as described in the ADS user manual.

Usage: `ALGO ADS ISTR` *option* `IOPT` *option* `IONE` *option* `IPRI` *option*

or: `ALGO IMSL`

Secondary Modifiers:

`ISTR` Strategy. Default is 8 – Sequential Quadratic Programming.

`IOPT` Optimizer. Default is 5 — Modified Method of Feasible Directions for constrained minimization.

`IONE` One dimensional search. Default is 7 – Determination of bounds followed by polynomial interpolation.

`IPRI` Print control. Default is 0 – No output printed. (Also recommended is 3120.)

`MINI` Specifies minimization

`MAXI` Specifies maximization.

Usage: `MINI/MAXI (EXPR) {Expression ID} WRT` *`list`*

`CONS` Constraint specification.

Usage: `CONS` *constraint$_1$ constraint$_2$* ...

The constraints have the forms:

```
X = Y
X .EQ. Y
X .LE. Y
X .LT. Y
X .GT. Y
X .GE. Y
```

where `X` and `Y` are constants, parameters or expressions (identified by the keyword `(EXPR)`).

```
| Examples:
| constraint u .le. 2
| constraint u .le. v
| constraint (expr) a .ge. 0
| constraint (expr)a .ge. (expr)b
```

Note that `.LE.` and `.LT.` are both interpreted as `.LE.` and that `.GE.` and `.GT.` are both interpreted as `.GE.`

Note also that multiple constraints may be combined in a single string.

```
| Example:
| constraint (expr) a .ge. 0
| constraint (expr)b .ge. (expr)a
| can be rewritten as:
| constraint (expr) b .ge. (expr) a .ge. 0
```

`BOUN` Boundary specification.

Usage: `BOUN` *bound$_1$ bound$_2$* ...

The bounds have the forms:

```
X = Y
```

```
X .EQ. Y
X .LE. Y
X .LT. Y
X .GT. Y
X .GE. Y
```

where `X` and `Y` are constants, parameters or expressions (identified by the keyword `(EXPR)`).

```
| Examples:
|
| bound u .le. 2
| bound u .le. v
| bound (expr) a .ge. 0
| bound (expr) a .ge. (expr) b
```

Note that `.LE.` and `.LT.` are both interpreted as `.LE.` and that `.GE.` and `.GT.` are both interpreted as `.GE.`.

Note also that multiple bounds may be combined in a single string.

```
| Example:
|
| bound (expr) a .ge. 0
| bound (expr) b .ge. (expr) a
|
| can be rewritten as:
|
| bound (expr) b .ge. (expr) a .ge. 0
```

`EXEC` Executes the optimization.

```
| Example (ADS test case):
|
| expr a : 2*sqrt(2)*X + Y
| expr b : (2*X + sqrt(2)*Y)/ (2*X*(X + sqrt(2)*Y))
| expr c : 1/(2*(X + sqrt(2)*Y))
|
| X = 1
| Y = 1
|
```

```
| opti init
| opti algo ads istrat 0 iopt 5 ioned 7
| minimize (expr) a wrt x y
| constraint (expr) b .le. 1
| constraint (expr) c .le. 1
| bound 0.01 .le. x .le. 1e20
| bound 0.01 .le. y .le. 1e20
|
| execute
|
| A = (expr) a
| B = (expr) b
| C = (expr) c
|
| display X Y
| display A B C
|
| The results follow:
|
| X : .7826923E+00 Y : .4152020E+00
| A : .2628990E+01 B : .1003817E+01 C : .3649964E+00
```

6.8.1 Optimization Notes

The gradient based algorithms called by this optimization module are reasonably robust when the functions being minimized or maximized and the imposed constraints are algebraic expressions, but problems can arise when the functions and constraints are generated by more complex analyses. Consider, for example, the following sequence of steps:

1) Part geometry defined by ANSYS for parameter vector (α_0, β_0, γ_0).
2) Geometry meshed in ANSYS and thermal/stress analysis executed
3) Volume V_0 calculated based on Step 2
4) Probabilistic fracture mechanics calculation set up and executed based on Step 2
5) Life L_0 corresponding to a 0.001 failure probability calculated based on Step 4
6) Part geometry defined by ANSYS for perturbed parameter vector (α_1, β_0, γ_0).
7) Steps 1 through 5 repeated for new geometry yielding volume V_1 and life L_1.

8) Partial derivative estimated: $\partial(V, L)/\partial\alpha = (V_1 - V_0, L_1 - L_0)/(\alpha_1 - \alpha_0)$

9) Steps repeated as necessary to estimate $\partial(V, L)/\partial\beta$ and $\partial(V, L)/\partial\gamma$.

The first problem concerns the estimation of the partials of V. The volume calculations for small perturbations of a model geometry can be compromised by differences in the automatically generated meshes. A small change in parameter α may yield a larger part volume in principle, but the differences in meshing between the two perturbations may yield a smaller estimated part volume. In this case, the estimate of $\partial V/\partial\alpha$ would have the wrong sign. While this can be overcome by iterative mesh refinement, doing so adds significant computational overhead.

The second problem affects the estimation of the partials of L and is more subtle. There is an element of statistical sampling in the probabilistic fracture mechanics calculations of Step 4. Sampling variations can overwhelm theoretical differences in failure probability between close geometric perturbations which in turn affects the estimate of $L_1 - L_0$. Again, brute force can be applied to overcome this problem: Convergence can be approached by increasing sampling in the relevant Monte Carlo algorithm. The penalty is longer execution time.

The following solution was felt to be better:

- A small region about the initial guess is defined. V and L are calculated at design points within and on the boundary of the region
- Local response surfaces $V(\alpha, \beta, \gamma)$ and $L(\alpha, \beta, \gamma)$ are fit to the calculations
- V is minimized in the region subject to the constraint that L remain above a specified value
- The solution defines a new local region, and the process continues until a convergence criterion is satisfied

This approach relies on larger perturbations in model geometry which are designed to result in larger (and less easily masked) differences in V and L; it also tends to average out the noise inherent in the meshing and lifing operations. Moreover, our approach for integrating geometric (and other) variability requires response surface modelling of risk and so benefits from the intermediate calculations.

6.9 Plotting

PDAS links into a powerful keyword–driven plotting library developed as part of SIESTA. Blocks of plotting commands are embedded into template programs by the `PLOT` primary keyword. An added PDAS capability: Lines can be defined by pointing to table columns.

```
| Example:
|
| tdat (table) 10 X Y Z
|
| line nlin 1 Y X
| line nlin 2 Z X
|
```

Produces two lines, one determined by the points (X_i, Y_i) defined by the the X and Y columns of table 10, the second by the points (X_i, Z_i) defined by the X and Z columns. If `TABL FLAG` is called prior to `TDAT`, only active rows of the table are considered.

6.10 Model, Mission and Calculation Libraries

Engines are broken down to components; components may be modeled in pieces – axisymmetric shells and quads for disk bores, webs, flanges, etc.; plane stress, plane strain quads or 3–D bricks for stress concentrated details such as bolt holes and dovetail slots. Each model may have several relevant missions (Air Combat, Air to Surface). And each mission may require more than one life calculation (multiple material assignments, differing sets of calculation points).

MISSYDD organizes models, missions and calculations, as well as material data, within a flexible database architecture. The program is driven by menus which are closely tied to the database index, and these menus are echoed by PDAS template commands.

There are many menu items, and many possible paths through an analysis. There are, however, eight basic steps:

1. Create a material library
   - Incubation models
   - Fracture mechanics constants
   - Inclusion distributions
2. Assemble a model
   - Geometric interfacing via SIESTA
   - Stress/thermal analysis interfacing via SIESTA
   - Material data assignments
   - Subsets
3. Define a mission
   - Load cases sequences
   - Scaled sequences
   - Superpositions
4. Initialize a computation
   - Material data assignments
   - Subsets
5. Initialize a life calculation
   - Incubation and fracture mechanics options
   - Crack area selection
6. Calculate fracture mechanics lives

7. Compute risk estimates
   - Fracture mechanics calculations and elemental volumes yield geometric failure distributions
   - Integrated with assigned defect distributions
   - Multiple model/mission/computations combined
8. Output
   - Elemental mission stresses/temperatures
   - Fracture mechanics lives
   - Risk estimates

6.11 General Commands

`SUBM` Submits user entered Unix commands.

Usage: `SUBM`
```
command
command
command
...
DONE
```

`DISP` Displays current values of specified parameters.

Usage: `DISP parameter parameter parameter ...`

`GLOB` Sets global options:

`QUIE` Suppresses echoing of template.

`ECHO` Enables echoing of template.

`(LIF` Sets the argument list for life calculations.

`EDIT` Edits a file, changing lines of the form:

parameter =

to:

parameter = current value

Usage: `EDIT`
```
filename
```

`WAIT` (Followed by `FILE`) Holds template execution until a specified file appears.

Usage: `WAIT FILE`
```
filename
```

7.0 Response Surfaces

Overlaid in Figure 7 are the failure distributions for nine design perturbations of a seeded model disk geometry. The significance is that these analyses (including geometry definition, finite element stress analysis, mission definition, property assignments, and probabilistic fracture mechanics analysis) were generated with a single PDAS template.

Summarizing:

- A response surface is defined to be a set of analysis nodes (geometry, thermal, stress, ...). Each node is tied to a vector of parameters (part dimensions, rotor speeds, ...), the variables of the analyses. Typically a response node involves a model (the subject), a mission (the action), and a computation (the result).
- A response surface is created by a template which is a set of keyword commands such as `MODL INIT` for Model Initialize, `MSSN COPY` for Mission Copy or `RISK EXEC` for Risk Execute. There are currently more than 200 distinct commands referring specifically to PDAS functions which can be used to build a template in addition to any other system recognized commands (Unix or program executables).
- The first step in execution generates copies of a master template, one for each response surface node included in the current run. Each copy is edited to name certain node–unique entries. For example, a parameter file (named `diskbase.prm`) used by ANSYS to define model geometry is copied into node–named files (`disk1.prm`, `disk2.prm`, `disk3.prm`, ...).
- PDAS cycles through each template copy, executing commands one at a time. If there is a hold up at some line, the program moves to the next node template. If all nodes are held up, the program goes to sleep. After predetermined length of time, it wakes up to check whether execution can resume.
- For example, the ANSYS parameter file `disk1.prm` is edited to assign parameter values (part dimensions, loading conditions). The geometry definition and finite element job is submitted via a distributed processing scheduler. At the end of the ANSYS execution, SIESTA functions are called to post–process the output file to create a SIESTA Random Data Base (RDB). PDAS looks for the node–named RDB files (`disk1.37` and `disk1.38`). Since the ANSYS job was just submitted, they will not be found, and PDAS proceeds onto the next node, editing `disk2.prm`, and so on.
- When `disk1.37` and `disk1.38` do show up, PDAS copies model, mission and computation formats, creating new MISSYDD database entries for response node 1. The new model entry references the RDB files for geometry information; it includes all required elemental material property assignments (incubation, fracture mechanics and inclusion distribution data). The new mission entry also references the RDB files; the mission is defined as a scaled sequence of load cases or superpositions of load cases. The computation entry includes specification of incubation and fracture mechanics options (e.g. gradients vs. no–gradients) and requested inclusion sizes.
- PDAS next performs fracture mechanics calculations yielding the life response of the model as a function of crack size, and integrates this response with the inclusion distribution to produce one of the failure distributions shown in Figure 7.

The PDAS architecture enables control of an enormous array of computations (it would have been as easy to execute 90 response surface nodes as it was to execute 9). While simple rules of logic must be obeyed in constructing the templates (computation definition must follow mission definition, which must follow model definition), there is great freedom in terms of organizing both internal and external program functions.

8.0 References

1. P.G. Roth, *Probabilistic Rotor Design System (PRDS) Phase 1*, WL–TR–92 –2011, Aero Propulsion and Power Directorate, Wright Laboratory, Air Force Systems Command, Wright–Patterson Air Force Base, Ohio 45433–6563, 1992.

2. C. (Raj) Sundararajan, editor, Probabilistic Structural Mechanics Handbook, Theory and Industrial Applications, Chapman & Hall, 1995.

3. B.M. Ayyub and R.H. McCuen, *Simulation–Based Reliability Methods*, Probabilistic Structural Mechanics Handbook, Theory and Industrial Applications, C. (Raj) Sundararajan, editor, Chapman & Hall, 1995, p. 53.

4. B. Carnahan, H.A. Luther and J.O. Wilkes, Applied Numerical Methods, John Wiley and Sons, Inc., 1969, p. 545.

5. D.H. Lehmer, *Mathematical Methods in Large Scale Computing Units*, Harvard Univ. Ann. Comput., 26, 1941, p. 141.

6. T. Stone, Demonstration of the periodicity associated with the MLC algorithm, private communication.

7. G. Marsaglia, A. Zaman and W.W. Tsang, Stat. Prob. Lett., 9, 1990, p. 35.

8. A. Haldar and S. Mahadevan, *First–Order and Second–Order Reliability Methods*, Probabilistic Structural Mechanics Handbook, Theory and Industrial Applications, C. (Raj) Sundararajan, editor, Chapman & Hall, 1995, p. 27.

9. NESSUS (Numerical Evaluation of Stochastic Structures Under Stress), Developed at the Southwest Research Institute, San Antonio, TX, 1991.

10. PROBAN (PROBability ANalysis), Developed at Det Norske Veritas through AS Veritas Research, Høvik, Norway, 1991.

11. CALREL (CAL–RELiability), Developed at the University of California at Berkeley by PL Liu, H.Z. Lin and A. Der Kiureghian, 1989.

12. ENSIP — USAF MIL–STD–1783.

13. P.G. Roth, *Validation of a Probabilistic Rotor Design System*, Proceedings of the 1992 ASME Winter Meeting, Anaheim, California, November 10, 1992, p. 255.

14. P.G. Roth, J.C. Murray, J.E. Morra and J.M. Hyzak, *Heavy Liquid Separation: A Reliable Method to Characterize Inclusions in Metal Powder*, Characterization, Testing and Quality Control, Advances in Powder Metallurgy & Particulate Materials – 1994, 2, Metal Powder Industries Federation, Princeton, NJ, 1994, p. 1.

15. W. Rudin, Real and Complex Analysis, 2'nd Ed., McGraw–Hill Series in Higher Mathematics, 1974, pp. 63–64.

16. C.F. Gerald and P.O. Wheatley, Applied Numerical Analysis, 3'rd Ed., Addison–Wesley, 1985, p. 537.

17. G.N. Vanderplaats, Numerical Optimization Techniques for Engineering Design: With Applications, McGraw–Hill, 1984.

18. ADS (Automated Design Synthesis), Developed by G.N. Vanderplaats under sponsorship of NASA and supported by Engineering Design Optimization, Inc., Santa Barbara, CA, 1987.

19. IMSL, Marketed by Visual Numerics, Inc., Houston, TX.

Appendix A

The following is the current Statement of Work for USAF Contract F33615–90–C–2070.

1.0 Introduction

The Integrated High Performance Turbine Engine Technology (IHPTET) Initiative has set the goal of doubling the performance of gas turbine engines by the year 2000. Achievement of this goal will place new demands on existing materials and will require the use of advanced material systems such as intermetallics and composites. Conventional, deterministic design methods are inherently conservative, and cannot be readily applied to composite material systems. Probabilistic design methodology offers a method of analyzing the interaction of the statistical distribution of design parameters. This reduces the conservatism of the design to a known, acceptable level, and gives the ability to deal with the complex interactions in composite material systems. This program will develop, validate and apply a probabilistic design system.

2.0 Scope

This program will consist of 6 phases. In Phase I, design and operating data will be collected and analyzed to form the basis of later work. In Phase II, the probabilistic design system will be developed, and in Phase III, the system will be validated through testing. Phase IV will apply the design system to a component which will be tested in Phase V. Finally, in Phase VI, the future application of the probabilistic design method will be explored, with particular reference to composite material systems.

3.0 Background

Conventional design systems are based on a deterministic approach, where a single value is assumed for each design factor. Safety is ensured by applying an arbitrary safety factor, or worst–case assumption. When several safety factors are combined in a design, the end result may be a very safe, but very conservative design. Achievement of IHPTET goals will require the use of new material systems and design concepts where it will be very difficult to determine satisfactory single values for design and safety factors. An alternative design system which has been developing in recent years is the probabilistic approach. In this approach, instead of single values, the statistical distributions of design factors are considered. The interaction of the distributions is analyzed, to predict an overall distribution of the life, or strength, of the total design. From the analysis, operating limitations or component life can be identified for a chosen probability of success. The main resultant benefits are weight reduction, through reduced design conservatism, and improved safety, through better definition of the probability of failure. The method can also be used to analyze an existing design to examine its sensitivity to the variability of influential factors.

4.0 Technical Requirements

The Contractor shall furnish all manpower, facilities and equipment to conduct the program. The Contractor shall exercise administrative and financial management functions during the course of this effort, including: scheduling of activities and milestones; describing status; outlining Contractor activity and progress towards accomplishment of objectives; planning, forecasting and making recommendations on funding and funding changes; program planning; describing in detail the overall results of the effort; and documenting any new technological breakthroughs

(CDRL Sequence No. 1, 2, 3, 5, 7, 8, 9). The program shall be conducted in 6 phases:

4.1 Phase I – Data Acquisition

In this phase, the Contractor shall collect, review and analyze data on rotor design, operation and failure, to provide a basis for the later phases of the program. Operating and failure data relating to compressor and turbine rotor disks shall be collected from published reports, laboratory experience, engine and component test results, and from field operations. The Contractor shall review and analyze the data to identify the factors which influence the strength and life of rotor disks and the variability of the influential factors. The failure modes of the disks shall also be determined.

4.1.1 Data Acquisition

The Contractor shall collect relevant data from laboratory, spin pit, test engine and field experience for evaluation and analysis. The Contractor shall compile the data from the Contractor's data bases, from available engine overhaul records, published reports, and other sources.

4.1.2 Correlation of Failure Experience

The Contractor shall identify the loadings, temperatures, and gradients that influence the structural longevity of turbine and compressor rotors. The Contractor shall further identify and quantify the probabilistic uncertainties associated with these variables. Potential failure modes shall also be identified.

4.1.3 Evaluation of Loads and Temperatures

The Contractor shall identify the loadings, temperatures, and gradients that influence the structural longevity of turbine and compressor rotors. The Contractor shall further identify and quantify the probabilistic uncertainties associated with these variables. Potential failure modes shall also be identified.

4.1.4 Probabilistic Rotor Design System Parameters

The Contractor shall identify the life–limiting parameters that place a structure at risk from the potential failure modes identified in Paragraph 4.1.3. The probabilistic influence of other risk concerns, such as manufacturing tolerances, imperfect non–destructive evaluation, and mission variability shall also be considered. The Contractor shall formulate an algorithmic architecture that will permit assessment of the overall probability of failure from competing failure modes, considering at least the primary, secondary, and tertiary threats.

4.1.5 Development of Acceptable Risk Criteria

The Contractor shall review available failure data. Data shall be sorted by failure mode at the component level whenever possible to identify specific trends in failure risk levels. Analytical trade studies shall be conducted using probabilistic methods to quantify actual failure risk levels for various peacetime and wartime scenarios. Study results shall be correlated to actual field experience whenever possible. The Contractor shall survey Government and industry sources to help define acceptable risk criteria based on current experience and on the results generated by the studies above. The results of Phase I shall be reviewed in an oral presentation to the Air Force (CDRL Sequence No. 4).

4.1.6 The Contractor shall obtain written approval from the Contracting Officer before proceeding with Phase II.

4.2 Phase II – Method Development

In this phase the Contractor shall review its existing probabilistic design methods and develop them as required for application to com-

pressor and turbine rotor disks. Models shall be selected or developed for all significant failure modes, including cyclic fatigue, fracture, burst, creep, vibration and growth. The models shall address the statistical variations of influential design factors, including mission profiles, material properties, inspection reliability and manufacturing capability. The Contractor shall also select and study an alternative model program, such as NESSUS (Numerical Evaluation of Stochastic Structures Under Stress), which shall be used later for comparison.

4.2.1 Failure Models Development

The Contractor shall assess existing probabilistic life cyclic fatigue and fracture models and modify them based on the results of Phase 1. The Contractor shall modify or develop probabilistic models for burst, creep, vibration, and growth failure modes.

4.2.2 Statistical Analysis Models

The Contractor shall construct the statistical models required as input to the analysis developed in task 4.2.1. These models shall include statistical variations of life drivers, regression models, and other analytical models. Statistical variation of life drivers shall include variations in the operating environment as well as the uncertainty in engineering analyses and models.

4.2.3 Methodology Description

The Contractor shall prepare a description of the analysis techniques contained in the code and the assumptions used or implicit in the methodology. At the completion of Phase II, the Contractor shall make an oral presentation to the Air Force (CDRL Sequence No. 4).

4.2.4 The Contractor shall obtain written approval from the Contracting Officer before proceeding with Phase III.

4.3 Phase III – Validation

In this phase the Contractor shall demonstrate the validity of the Probabilistic Rotor Design System (PRDS) design tool. This shall be done by designing specimens, and testing them to failure in a series of controlled experiments. All test hardware shall be designed, fabricated and tested in accordance with AFR 800–16 and the standards of MIL–STD–882B. The specimens shall be designed to simulate the failure behavior of a turbine rotor disk, and the testing shall aim to produce representative failures in the specimens. The specimen design shall be analyzed using the probabilistic design method and the probability distribution of its failure shall be determined. The test plan shall be designed to produce a statistically significant validation of the predicted results, using a design of experiments methodology.

4.3.1 Test Definition

The Contractor shall determine the test articles and necessary tests to be used for validating the PRDS. The test articles and test methods shall be designed so as to provide a clear validation of the probabilistic life methodology. The Contractor shall perform both a deterministic and probabilistic life analysis of the test articles. The Contractor shall then determine the tests required to verify the predictions and their applicability to future testing (CDRL Sequence No. 6). The Contractor shall review and approve the fabrication of the specialized tooling required to cyclic spin selected test articles in a government facility.

4.3.2 Test Evaluation

The Contractor shall fabricate the approved test articles and monitor the PRDS validation testing in accordance with the approved test plan. The contractor shall monitor cyclic spin testing of selected components until each has failed or until economic life has been attained. Fractographic examination of selected test articles

will be conducted to characterize the location (surface, near–surface, subsurface, etc.) and nature (ceramic, oxide, nitride, void, crystallographic, etc.) of the origin of the fracture. All raw test data shall be compiled (CDRL Sequence No. 10) and the Contractor shall make an oral presentation to the Air Force (CDRL Sequence No. 4).

4.3.3 PRDS Design Tool Refinement

Based on the results of the validation tests the Contractor shall make improvements to the PRDS design tool as required.

4.3.4 The Contractor shall obtain written approval from the Contracting Officer before proceeding with Phase IV.

4.4 Phase IV – Application

In this phase the Contractor shall apply the Probabilistic Rotor Design System (PRDS) to one turbine rotor disk from an advanced technology engine. The selected component shall be used for a case study addressing the use of the PRDS design tool as a means of reducing weight. Representative engine cycle parameters (stress and temperature histories) and available materials data will be identified and analyzed with the refined methodology resulting from Phase II, to establish a baseline weight and survival probability of the selected components for a specifically defined life.

4.4.1 Probabilistic Redesign

The Contractor shall redesign and analyze the component for the same life as the baseline design using the PRDS code, in sufficient detail to determine the survival probability and weight for comparison to the baseline design. The objective of this task is to identify an alternative design that provides a significant weight reduction as compared to the baseline design, with an acceptable survival probability identified in Phase I. At the completion of Phase IV, the results shall be reviewed with the Air Force (CDRL Sequence No. 4).

4.4.2 The Contractor shall obtain written approval from the Contracting Officer before proceeding with Phase V.

4.5 Phase V – Application Test

In this phase the Contractor shall monitor testing of the rotor disk which was redesigned in Phase IV. The disk shall be provided from a separately funded contract as GFE. Under a separately funded contract, the Contractor shall install the disk in a test engine to be run for a predetermined number of cycles. After the engine test, the Contractor shall monitor the disk installation in a government facility spin rig and monitor testing of the disk to failure. Failure analysis shall be conducted and all test results shall be evaluated to demonstrate the integrity of the design.

4.5.1 Rotor Components Detail Design

The Contractor shall establish and document the design details of the selected rotor disk using the Probabilistic Rotor Design System (PRDS). The acceptable risk criteria defined during Phase I shall be integrated with probabilistic methods to establish the design allowables. These allowables shall then be used in conjunction with standard design analysis procedures such as finite element analysis to establish the final component. This detail design shall include the statistical probability of failure for all primary failure modes. The Contractor shall also evaluate the weight and performance payoff obtained by redesigning the component using probabilistic design concepts.

4.5.2 Rig Test and Validation for Gas Generator Testing

The Contractor shall monitor and analyze the test of the GFE rotor disk to provide final verification of the PRDS and applicability for advanced demonstrator engine testing. The Contractor shall: (1) prepare a test plan defining post–engine component testing, (2) review and approve design and fabrication of necessary test rig adaptive hardware, (3) perform a complete stress survey. The Contractor shall identify and document all instrumentation requirements. The Contractor shall identify the design conditions and cycles to be simulated. The Contractor shall define evaluation methods for analysis and test correlation. The Contracting Officer, within 7 days after submittal of the test plan (CDRL Sequence No. 6), will inform the Contractor by letter if work may proceed on the Phase V test items. Following approval by the Air Force Contracting Officer, the Contractor shall provide the PRDS component for testing in a separately funded advanced demonstrator engine test. The Contractor shall conduct engine testing in such a manner as to obtain data for verification of the design predictions and the applicability of advanced PRDS components in advanced turbopropulsion weapon systems.

4.5.3 Hot Cyclic Spin of Component

The Contractor shall monitor a hot cyclic spin test of the GFE rotor disk until it fails. This test shall be completed after all potential advanced gas generator testing has occurred. The Contractor shall conduct a post–test failure analysis of the rotor component. The results shall be compared against predictions from the Probabilistic Rotor Design System to determine the accuracy of the system. The Contractor shall review the results of Phase V in an oral presentation to the Air Force (CDRL Sequence No. 4).

4.5.4 The Contractor shall obtain written approval from the Contracting Officer before proceeding with Phase VI.

4.6 Phase VI – Method Extension

In this phase the Contractor shall examine the extension of the probabilistic design method to other component types and materials. Advanced materials, and engine components other than rotor disks, shall be classified according to their properties, manufacturing methods and failure modes, and the applicability of the probabilistic design system shall be determined. Recommendations shall be made for further work, including modification or extension of the design system.

4.6.1 PRDS for Metal Matrix Composite (MMC) Rotors

The Contractor shall gather information on the properties, behavior and failure modes for MMC materials. This information shall be evaluated to determine the possibilities and requirements for developing statistical models for a design system. The Contractor shall determine the benefits of a Probabilistic Rotor Design System for meeting MIL–STD–1783 Engine Structural Integrity Program (ENSIP) and IHPTET Requirements for MMC materials.

4.6.2 Modifications to Military Standard 1783

The Contractor shall develop a modified Military Standard 1783 that incorporates PRDS. The modification shall be in a form that can be submitted to the Government Committee on Military Standards for consideration.

4.6.3 PRDS Modifications

The probabilistic life–prediction methodology, resulting from Phase II, shall be modified or supplemented as applicable to address the failure mechanisms determined by Paragraph

4.6.1, and the information obtained in Phases III, IV and V. A plan addressing the comprehensive application of the PRDS methodology shall be prepared. The plan shall be developed to include both isotropic and anisotropic materials. The results of Phase VI shall be reviewed in an oral presentation to the Air Force (CDRL Sequence No. 4).

5.2 The Contractor shall plan for six status reviews in the form of oral presentations to the Air Force at Wright–Patterson Air Force Base (CDRL Sequence No. 4).

Appendix B

The scatter inherent in small scale sampling is significant and absolutely unavoidable. Repeat samples of size 20 could be expected to yield distributions as shown in Figures B.1 through B.4. (A count somewhere on the order of 20 inclusions is not untypical for a single half–pound HLS sample.)

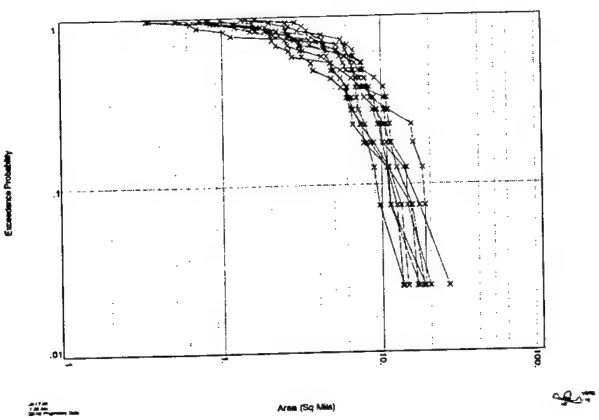


Figure B.1. Reproducibility – 10 Random Samples of Size 20.

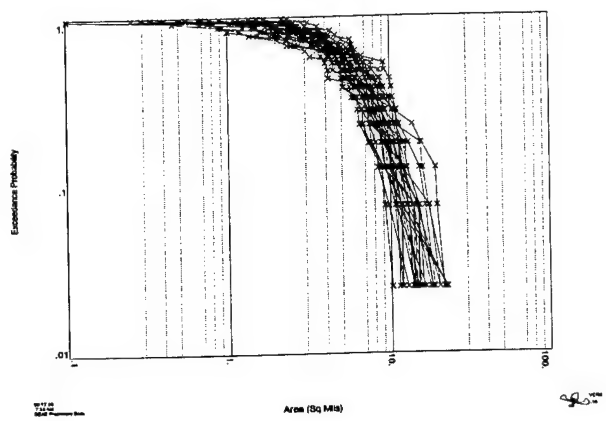


Figure B.3. Reproducibility – 30 Random Samples of Size 20.

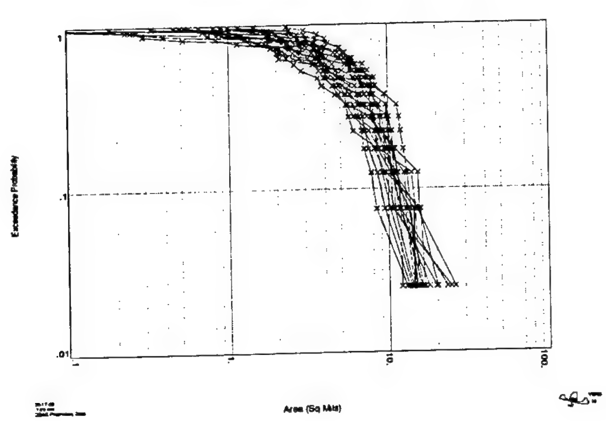


Figure B.2. Reproducibility – 20 Random Samples of Size 20.

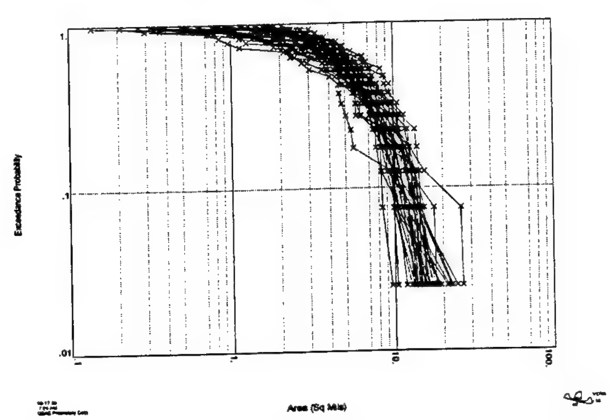


Figure B.4. Reproducibility – 40 Random Samples of Size 20.

Appendix C

The following templates were used to produce the analyses presented in Section 2.1.

Direct Integration

```
table 1 initialize                                         Define X–distribution
table 1 column Z from -6 to 6 # 1000
table 1 set Z = Z + 0
table 1 set PROB = (1/sqrt(2*pi))*exp(-((Z-0)**2)/2)

distribution input X (table) 1 Z (format) density

table 1 initialize                                         Define Y–distribution
table 1 column Z from -6 to 6 # 1000
table 1 set Z = Z + 0
table 1 set PROB = (1/sqrt(2*pi))*exp(-((Z-3)**2)/2)

distribution input Y (table) 1 Z (format) density

table 2 initialize                                         Construct table of V–values
V = 0.7
table 2 write new# V
V = 1.3
table 2 write new# V
V = 2.0
table 2 write new# V

loop start                                                 Loop over the V–table
loop over (table) 2

expr B : (X+V)**2 + (X-V)**2 - 2                           Define expressions used to compute
expr C : (X**2-v**2)**2 - (X+V)**2 - (X-V)**2              Y–limits of integration
expr DSCR : B**2 - 4*C

table 1 initialize (width) 30                              Construct table of X–values
table 1 column X from -3.3 to 3.3 # 1000

table 1 set B = (expr) B                                   Determine Y–limits of integration
table 1 set C = (expr) C                                   for each X
table 1 set DSCR = (expr) DSCR
```

```
table 1 flag DSCR .ge. 0
table 1 set YSQ = (-B + sqrt(DSCR))/2
table 1 flag YSQ .ge. 0
table 1 set YLIM = sqrt(YSQ)
```

Loop over the X–table

```
loop start
loop over (table) 1
```

Define Y–integrand as 1 between ±`YLIM` and 0 elsewhere

```
expr : unsf(YLIM - Y) * unsf(YLIM + Y)
algebra replace YLIM
```

Evaluate the Y–integral and write to the X–table

```
Y_INT = (integral) LAST wrt Y
table 1 writ Y_INT
```

End of the X–table loop

```
loop stop
```

Define X–integrand as the X–digitized function of the Y–integral

```
expr : (dig) (table) 1 Y_INT (lin) X (lin)
```

Integrate this function; Write to the V–table. (`X_INT` estimates failure probability)

```
X_INT = (integral) LAST wrt X
table 2 write X_INT
```

End of the V–table loop

```
loop stop
```

Remove the file `example.1.out` (if it exists)

```
submit
rm example.1.out
done
```

Output V and `X_INT` to `example.1.out`

```
table 2 output V X_INT
example.1.out
```

Monte Carlo Integration

Define X–distribution; First the probability density function

```
table 1 initialize
table 1 column Z from -6 to 6 # 1000
table 1 set Z = Z + 0
table 1 set PROB = (1/sqrt(2*pi))*exp(-((Z-0)**2)/2)
```

Then, integrate to yield the cumulative distribution function

```
S = 0
table 1 set S = S + PROB
```

```
table 1 read row# 1000
SUM = S
table 1 set PROB = S/SUM

dist input X (table) 1 Z (format) cumulative

table 1 initialize                                          Define Y-distribution
table 1 column Z from -6 to 6 # 1000                        First the probability density
table 1 set Z = Z + 3                                       function
table 1 set PROB = (1/sqrt(2*pi))*exp(-((Z-3)**2)/2)
S = 0                                                       Then, integrate to yield the
table 1 set S = S + PROB                                    cumulative distribution function
table 1 read row# 1000
SUM = S
table 1 set PROB = S/SUM

dist input Y (table) 1 Z (format) cumulative

table 2 initialize                                          Construct table of V-values
V = 0.7
table 2 write new# V
V = 1.3
table 2 write new# V
V = 2.0
table 2 write new# V

loop start                                                  Loop over the V-table
loop over (table) 2

table 1 init (width) 15                                     Construct table of 1,000,000
table 1 rand X Y # 1000000                                  randomly chosen (X, Y) points

table 1 set DEN1 = (X - V)**2 + Y**2                        Compute pressure at each point
table 1 set DEN2 = (X + V)**2 + Y**2
table 1 set P = 1e10
table 1 flag DEN1 .gt. 0 DEN2 .gt. 0
table 1 set P = 1/DEN1 + 1/DEN2

table 1 flag P .ge. 1                                       Flag points with pressure greater
table 1 set COUNT = 1                                       than or equal to 1; Count the rows
SUM = (table) 1 (sum) COUNT                                 and divide by 1,000,000; Write to
```

```
FRACTION = SUM/1000000
table 2 write FRACTION
```

the V–table. (FRACTION estimates failure probability)

```
loop stop
```

End of the V–table loop

```
submit
rm example.2.out
done
```

Remove the file example.2.out (if it exists)

```
table 2 output V FRACTION
example.2.out
```

Output V and FRACTION to example.2.out

First Order Reliability Approximation

```
table 1 initialize
table 1 column Z from -6 to 6 # 1000
table 1 set Z = Z + 3
table 1 set PROB = (1/sqrt(2*pi))*exp(-((Z-3)**2)/2)

distribution input Z (table) 1 Z (format) density
```

Define Z–distribution

```
table 2 initialize
V = 0.7
table 2 write new# V
V = 1.3
table 2 write new# V
V = 2.0
table 2 write new# V
```

Construct table of V–values

```
loop start
loop over (table) 2
```

Loop over the V–table

```
expr B : (X+V)**2 + (X-V)**2 - 2
expr C : (X**2-V**2)**2-(X+V)**2-(X-V)**2
expr DSCR : B**2 - 4*C
```

Define expressions used to compute Y–bounds

```
table 1 initialize (width) 30
table 1 column X from -3.3 to 3.3 # 1000
```

Construct table of X–values

```
table 1 set B = (expr) B
table 1 set C = (expr) C
```

For each X, determine Y–bound

```
table 1 set DSCR = (expr) DSCR
table 1 flag DSCR .ge. 0
table 1 set YSQ = (-B + sqrt(DSCR))/2
table 1 flag YSQ .ge. 0
table 1 set YBND = sqrt(YSQ)
```

`YBND = (table) 1 (max) YBND`	Determine maximum Y–bound
`expr : unsf(-Z + YBND)` `algebra replace YBND`	Define Z–integrand as 1 below `YBND` and 0 above
`Z_INT = (integral) LAST wrt z` `table 2 write X_INT`	Integrate this function; Write to the V–table. (`Z_INT` estimates failure probability)
`loop stop`	End of the V–table loop

`submit`
`rm example.3.out`
`done`
	Remove the file `example.3.out` (if it exists)

`table 2 output V X_INT` `example.3.out`	Output V and `Z_INT` to `example.3.out`